aws

Guida per l'utente

AWS Identity and Access Management

AWS Identity and Access Management: Guida per l'utente

Table of Contents

Che cos'è IAM?

Follow us on Twitter

AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse. Con IAM, puoi gestire centralmente le autorizzazioni che controllano le AWS risorse a cui gli utenti possono accedere. Utilizza IAM per controllare chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) per l'utilizzo di risorse.

Quando crei un Account AWS account, inizi con un'unica identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.

Indice

Video introduttivo a IAM

AWS Training and Certification offre un video introduttivo di 10 minuti a IAM:

Introduzione a AWS Identity and Access Management

Caratteristiche di IAM

IAM offre le seguenti funzionalità:

Accesso condiviso al tuo Account AWS

Puoi concedere ad altri utenti le autorizzazioni per amministrare e utilizzare le risorse nel tuo account AWS senza la necessità di condividere la password o la chiave di accesso.

Autorizzazioni granulari

Puoi concedere autorizzazioni diverse a diverse persone per diverse risorse. Ad esempio, potresti consentire ad alcuni utenti l'accesso completo ad Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift e altri servizi. AWS Per gli altri utenti, puoi consentire l'accesso in sola lettura ad alcuni bucket S3, oppure l'autorizzazione ad amministrare solo alcune istanze EC2 oppure l'autorizzazione ad accedere alle informazioni di fatturazione, ma nient'altro.

Accesso sicuro alle AWS risorse per le applicazioni eseguite su Amazon EC2

Puoi utilizzare le funzionalità di IAM per fornire in maniera sicura le credenziali per le applicazioni che funzionano su istanze EC2. Queste credenziali forniscono all'applicazione le autorizzazioni per accedere ad altre risorse. AWS Alcuni esempi includono i bucket S3 e le tabelle DynamoDB.

Autenticazione a più fattori (MFA)

Puoi aggiungere l'autenticazione a due fattori per il tuo account e per i singoli utenti per maggiore sicurezza. Con MFA tu o i tuoi utenti dovete fornire non solo una password o la chiave di accesso che funzioni con il tuo account, ma anche un codice da un dispositivo appositamente configurato. Se utilizzi già una chiave di sicurezza FIDO con altri servizi e ha una configurazione AWS supportata, puoi utilizzarla WebAuthn per la sicurezza MFA. Per ulteriori informazioni, consulta Configurazioni supportate per l'utilizzo di chiavi di accesso e chiavi di sicurezza.

Federazione delle identità

Puoi consentire agli utenti che utilizzano già le password altrove, ad esempio nella tua rete aziendale o con un provider di identità Internet, di ottenere l'accesso temporaneo al tuo Account AWS.

Informazioni d'identità per la sicurezza

Se utilizzi AWS CloudTrail riceverai i record del log che includono le informazioni su chi effettua le richieste per le risorse nel tuo account. Queste informazioni sono basate sulle identità IAM.

Conformità PCI DSS

IAM supporta l'elaborazione, l'archiviazione e la trasmissione di dati di carte di credito da parte di un esercente o di un provider di servizi, oltre a essere conforme allo standard Payment Card Industry Data Security Standard (PCI DSS). Per ulteriori informazioni su PCI DSS, incluso come richiedere una copia del PCI AWS Compliance Package, vedere PCI DSS Level 1.

Integrato con molti servizi AWS

Per un elenco di AWS servizi compatibili con IAM, consultaAWS servizi che funzionano con IAM.

Consistente finale

IAM, come molti altri AWS servizi, alla fine è coerente. IAM raggiunge un'alta disponibilità replicando i dati su più server nei data center di Amazon di tutto il mondo. Se una richiesta per modificare alcuni dati ha successo, la modifica viene completata e memorizzata in maniera sicura. Tuttavia, le modifiche devono essere replicate su IAM e questo può richiedere tempo. Tali modifiche includono la creazione o l'aggiornamento di utenti, gruppi, ruoli, o policy. Si consiglia di non includere tali modifiche IAM nei percorsi critici e ad alta disponibilità del codice dell'applicazione. Al contrario, apporta modifiche IAM in un'inizializzazione separata o in una routine di configurazione che si esegue meno frequentemente. Inoltre, assicurarsi di verificare che le modifiche siano state propagate prima che i flussi di lavoro di produzione dipendano da esse. Per ulteriori informazioni, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

Utilizzo gratis

AWS Identity and Access Management (IAM) e AWS Security Token Service (AWS STS) sono funzionalità del tuo AWS account offerte senza costi aggiuntivi. Ti viene addebitato solo quando accedi ad altri AWS servizi utilizzando gli utenti IAM o le credenziali di sicurezza AWS STS temporanee. Per informazioni sui prezzi di altri AWS prodotti, consulta la pagina dei prezzi di Amazon Web Services.

Accesso a IAM

Puoi lavorare con AWS Identity and Access Management in uno dei seguenti modi.

AWS Management Console

La console è un'interfaccia basata su browser per gestire IAM e AWS risorse. Per ulteriori informazioni sull'accesso a IAM tramite la console, consulta Come accedere ad AWS nella Guida per l'utente di Accedi ad AWS .

AWS Strumenti da riga di comando

È possibile utilizzare gli strumenti della riga di AWS comando per impartire comandi dalla riga di comando del sistema per eseguire IAM e AWS attività. L'utilizzo della riga di comando può essere più veloce e semplice rispetto all'uso della console. Gli strumenti da riga di comando sono utili anche se desideri creare script che eseguano AWS attività.

AWS fornisce due set di strumenti da riga di comando: the AWS Command Line Interface(AWS CLI) e the AWS Tools for Windows PowerShell. Per informazioni sull'installazione e l'utilizzo di AWS CLI, consulta la Guida AWS Command Line Interface per l'utente. Per informazioni sull'installazione e l'utilizzo degli strumenti per Windows PowerShell, consulta la Guida per AWS Tools for Windows PowerShell l'utente.

Dopo aver effettuato l'accesso alla console, puoi utilizzarla AWS CloudShell dal tuo browser per eseguire i comandi CLI o SDK. Le autorizzazioni per l'accesso alle AWS risorse si basano sulle credenziali utilizzate per accedere alla console. A seconda della tua esperienza, potresti ritenere che la CLI sia un metodo più efficiente per gestire il tuo Account AWS. Per ulteriori informazioni, consulta Utilizzo AWS CloudShell per l'utilizzo con AWS Identity and Access Management

AWS SDK

AWS fornisce SDK (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby, .NET, iOS, Android, ecc.). Gli SDK offrono un modo conveniente per creare un accesso programmatico a IAM e. AWS Ad esempio, gli SDK si occupano di attività quali la firma crittografica delle richieste, la gestione degli errori e la ripetizione automatica delle richieste. Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta la pagina Tools for Amazon Web Services.

API Query IAM

Puoi accedere a IAM e in modo AWS programmatico utilizzando l'API IAM Query, che consente di inviare richieste HTTPS direttamente al servizio. Quando utilizzi l'API Query, devi

includere il codice per firmare in modo digitale le richieste utilizzando le tue credenziali. Per ulteriori informazioni, consulta Chiamata all'API IAM utilizzando le richieste di query HTTP e Documentazione di riferimento dell'API IAM.

Quando si usa IAM?

Quando si eseguono diverse funzioni lavorative

AWS Identity and Access Management è un servizio di infrastruttura di base che fornisce le basi per il controllo degli accessi basato sulle identità interne. AWS IAM viene utilizzato ogni volta che accedi al tuo account AWS .

Le modalità di utilizzo di IAM cambiano in base alle operazioni eseguite in AWS.

- Utente del servizio: se utilizzi un servizio AWS per eseguire il tuo lavoro, allora l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità utilizzate per il lavoro, potrebbero essere necessarie altre autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore.
- Amministratore del servizio: se sei responsabile di una AWS risorsa presso la tua azienda, probabilmente hai pieno accesso a IAM. Il tuo compito è determinare le funzionalità e le risorse IAM a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM.
- Amministratore IAM: se sei un amministratore IAM, puoi gestire le identità IAM e scrivere policy per gestire l'accesso ad IAM.

Quando vieni autorizzato ad accedere alle risorse AWS

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle

identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

Quando accedi come utente IAM

Un utente IAM è un'identità interna a te Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente di IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali

temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente di IAM.

Quando assumi un ruolo IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

  - Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente di IAM.

Quando crei policy e autorizzazioni

Concedi le autorizzazioni a un utente creando una policy che è un documento che elenca le operazioni che un utente può eseguire e le risorse che tali operazioni possono influenzare. Qualsiasi operazione o risorsa che non è esplicitamente consentita viene negata come impostazione

predefinita. Le policy possono essere create e collegate ai principali (utenti, gruppi di utenti, ruoli assunti da utenti e risorse).

Queste policy vengono utilizzate con un ruolo IAM:

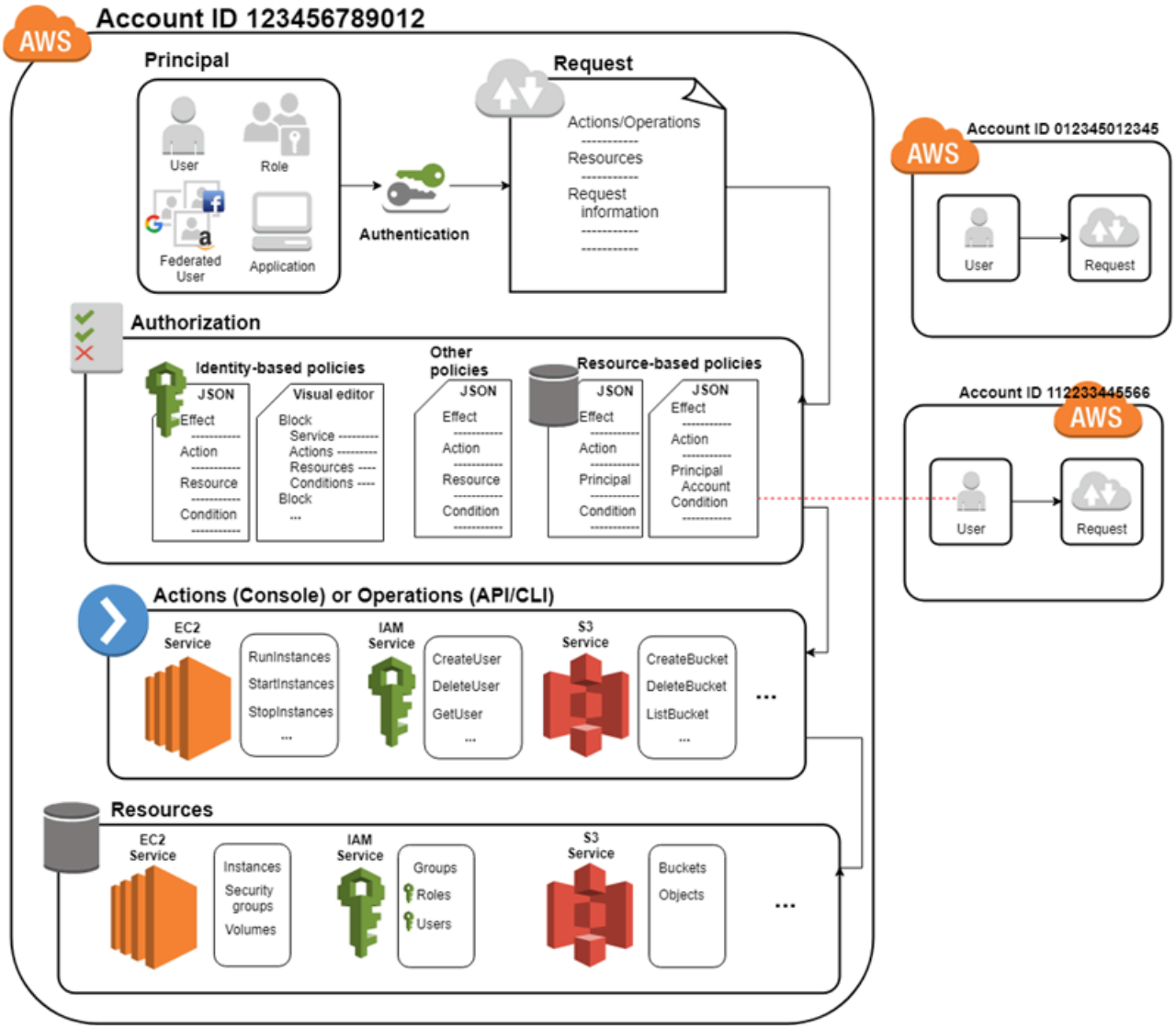
- Policy di attendibilità: definisce quali principali possono assumere il ruolo e a quali condizioni. Una policy di attendibilità è un tipo specifico di policy basata sulle risorse per i ruoli IAM. Un ruolo può avere una sola policy di attendibilità.
- Policy basate sull'identità (in linea e gestite): queste policy definiscono le autorizzazioni che l'utente del ruolo è in grado di eseguire (o che non può eseguire) e su quali risorse.

Utilizza gli Esempi di policy basate su identità IAM per definire le autorizzazioni per le identità IAM. Una volta trovata la policy desiderata, seleziona view the policy (visualizza la policy) per consultare il JSON della policy. Puoi utilizzare il documento di policy JSON come modello per le tue policy.

> **Note**
>
> Se utilizzi il Centro identità IAM per gestire i tuoi utenti, assegni set di autorizzazioni nel Centro identità IAM invece di collegare una policy di autorizzazioni a un principale. Quando si assegna un set di autorizzazioni a un gruppo o utente in Centro identità AWS IAM, IAM Identity Center crea i ruoli IAM corrispondenti in ciascun account e associa le politiche specificate nel set di autorizzazioni a tali ruoli. Il Centro identità IAM gestisce il ruolo e consente agli utenti autorizzati che hai definito di assumerlo. Se modifichi il set di autorizzazioni, il Centro identità IAM garantisce che le policy e i ruoli IAM corrispondenti vengano aggiornati di conseguenza.
> Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Funzionamento di IAM

IAM offre l'infrastruttura necessaria per gestire l'autenticazione e l'autorizzazione per il tuo Account AWS. Il diagramma seguente illustra tale infrastruttura IAM:



Innanzitutto, per autenticarsi con AWS, un utente umano o un'applicazione utilizza le proprie credenziali di accesso. L'autenticazione viene fornita associando le credenziali di accesso a un soggetto principale (un utente IAM, un utente federato, un ruolo IAM o un'applicazione) considerato affidabile da Account AWS.

Successivamente, viene fatta una richiesta per concedere al principale accesso alle risorse. L'accesso è concesso in risposta a una richiesta di autorizzazione. Ad esempio, quando accedi per la prima volta alla console e ti trovi nella home page, non stai accedendo a un servizio specifico. Quando selezioni un servizio, la richiesta di autorizzazione viene inviata a quel servizio e verifica se la tua identità è nell'elenco degli utenti autorizzati, quali policy vengono applicate per controllare il livello di accesso concesso e qualsiasi altra policy che potrebbe essere in vigore. Le richieste di

autorizzazione possono essere fatte dai responsabili interni all'azienda Account AWS o da terzi di cui ci si fida. Account AWS

Una volta autorizzato, il principale può intervenire o eseguire operazioni sulle risorse del tuo Account AWS. Ad esempio, il principale potrebbe avviare una nuova Amazon Elastic Compute Cloud istanza, modificare l'appartenenza al gruppo IAM o eliminare i Amazon Simple Storage Service bucket.

Concetti di base

- Termini
- Principale
- Richiesta
- Autenticazione
- Autorizzazione
- Azioni o operazioni
- Risorse

Termini

Questi termini IAM sono comunemente usati quando si lavora con AWS:

Risorse IAM

Le risorse IAM sono archiviate in IAM. Puoi aggiungerle, modificarle e rimuoverle da IAM.

- Utente
- gruppo
- role
- policy
- oggetto identity-provider

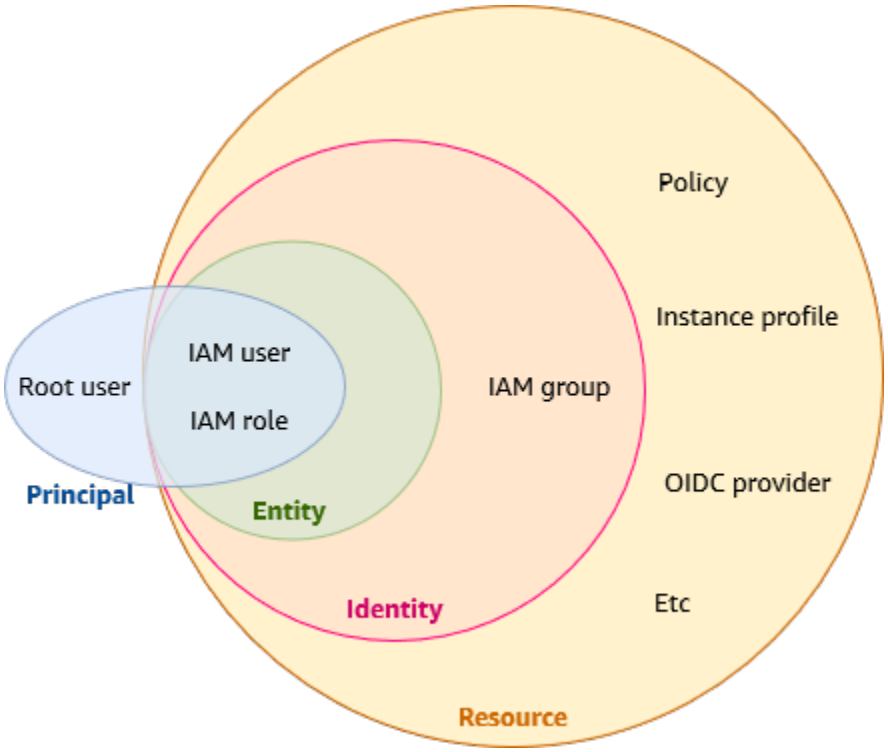
Entità IAM

Risorse IAM AWS utilizzate per l'autenticazione. Le entità possono essere specificate come Principale in un criterio basato sulle risorse.

- Utente
- role

Identità IAM

Una risorsa IAM che può essere autorizzata nelle policy per eseguire azioni e accedere alle risorse. Le identità includono utenti, gruppi e ruoli.



Principali

Una persona o un'applicazione che utilizza il Utente root dell'account AWS, un utente IAM o un ruolo IAM a cui accedere ed effettuare richieste AWS. Questi includono utenti federati e ruoli assunti.

Utenti

Noti anche come identità umane, gli utenti sono le persone, gli amministratori, gli sviluppatori, gli operatori e i consumer delle tue applicazioni.

Carico di lavoro

Una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione o un processo back-end. Può includere applicazioni, strumenti operativi e componenti.

Principale

Un principale è un utente o un carico di lavoro umano che può richiedere un'azione o un'operazione su una AWS risorsa. Dopo l'autenticazione, al principale possono essere concesse credenziali permanenti o temporanee a cui effettuare richieste AWS, a seconda del tipo principale. Agli utenti IAM e all'utente root vengono concesse credenziali permanenti, mentre ai ruoli vengono concesse credenziali temporanee. Come procedura ottimale, si consiglia di richiedere agli utenti umani e ai carichi di lavoro di accedere alle AWS risorse utilizzando credenziali temporanee.

Richiesta

Quando un principale tenta di utilizzare l' AWS Management Console, l' AWS API o il AWS CLI, quel principale invia una richiesta a. AWS La richiesta include le informazioni seguenti:

- Azioni o operazioni: le operazioni che l'entità principale vuole eseguire. Può trattarsi di un'azione nell' AWS Management Console o di un'operazione nell' AWS API AWS CLI o.
- Risorse: l'oggetto AWS risorsa su cui vengono eseguite le azioni o le operazioni.
- Principale – Persona o applicazione che utilizza un'entità (utente o ruolo) per inviare la richiesta. Le informazioni sul principale includono le policy associate all'entità che il principale ha utilizzato per accedere.
- Dati di ambiente: informazioni sull'indirizzo IP, l'agente utente, lo stato SSL abilitato o l'ora del giorno.
- Dati sulla risorsa – I dati correlati alla risorsa che viene richiesta. Possono essere incluse informazioni quali un nome di tabella di DynamoDB o un tag su un'istanza Amazon EC2.

AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta, che viene utilizzato per valutare e autorizzare la richiesta.

Autenticazione

Un principale deve essere autenticato (effettuato l'accesso AWS) utilizzando le proprie credenziali a cui inviare una richiesta. AWS Alcuni servizi, come Amazon S3 e AWS STS, consentono alcune richieste da parte di utenti anonimi. Tuttavia, si tratta di eccezioni alla regola.

Per eseguire l'autenticazione dalla console come utente root, devi effettuare l'accesso con indirizzo e-mail e password. In qualità di utente federato, sei autenticato dal tuo provider di identità e ti viene

concesso l'accesso alle AWS risorse assumendo ruoli IAM. In qualità di utente IAM, fornisci il tuo ID account o alias e quindi il nome utente e la password. Per autenticare i carichi di lavoro dall'API oppure dalla AWS CLI, potresti utilizzare credenziali temporanee assegnandoti un ruolo oppure potresti utilizzare credenziali a lungo termine fornendo la tua chiave di accesso e la chiave segreta. Potrebbe inoltre essere necessario fornire ulteriori informazioni di sicurezza. Come best practice, ti consigliamo AWS di utilizzare l'autenticazione a più fattori (MFA) e credenziali temporanee per aumentare la sicurezza del tuo account. Per ulteriori informazioni sulle entità IAM che AWS possono autenticarsi, consulta e. Utenti IAM Ruoli IAM

Autorizzazione

È inoltre necessario essere autorizzati (consentiti) per completare la richiesta. Durante l'autorizzazione, AWS usa i valori del contesto della richiesta per cercare policy applicabili alla richiesta. Quindi, utilizza le policy per determinare se accettare o rifiutare la richiesta. La maggior parte delle policy viene archiviata AWS come documenti JSON e specifica le autorizzazioni per le entità principali. Vi sono diversi tipi di policy che possono influire sull'autorizzazione di una richiesta. Per fornire agli utenti le autorizzazioni per accedere alle AWS risorse del proprio account, sono necessarie solo politiche basate sull'identità. Le policy basate su risorse sono molto utilizzate per concedere l'accesso a più account. Gli altri tipi di policy sono caratteristiche avanzate e vanno usate con cautela.

AWS controlla ogni politica che si applica al contesto della richiesta. Se una singola politica di autorizzazioni include un'azione negata, AWS nega l'intera richiesta e interrompe la valutazione. Questa azione si chiama rifiuto esplicito. Poiché le richieste vengono rifiutate per impostazione predefinita, AWS autorizza la richiesta solo se ogni parte della richiesta è consentita dalle politiche di autorizzazione applicabili. La logica di valutazione per una richiesta all'interno di un singolo account segue queste regole generali:

- Come impostazione predefinita, tutte le richieste vengono negate. (In generale, le richieste effettuate utilizzando le credenziali Utente root dell'account AWS per risorse nell'account sono sempre consentite).
- Un'autorizzazione esplicita in una policy di autorizzazione qualsiasi (basata su identità o basata su risorse) sostituisce questa impostazione predefinita.
- L'esistenza di un SCP Organizations, un limite delle autorizzazioni IAM o una policy di sessione sostituisce l'autorizzazione. Se esiste uno o più di questi tipi di policy, devono tutti consentire la richiesta. In caso contrario, viene rifiutata implicitamente.
- Un rifiuto esplicito in una policy sostituisce qualsiasi permesso.

Per ulteriori informazioni su come vengono valutati tutti i tipi di policy, consulta Logica di valutazione delle policy. Se devi effettuare una richiesta in un altro account, una policy nell'altro account deve consentirti di accedere alla risorsa e l'entità IAM che utilizzi per effettuare la richiesta deve avere una policy basata su identità che consenta la richiesta.

Azioni o operazioni

Dopo che la richiesta è stata autenticata e autorizzata, AWS approva le azioni o le operazioni contenute nella richiesta. Le operazioni vengono definite da un servizio e includono le azioni che puoi effettuare su una risorsa, come la visualizzazione, la creazione, la modifica e l'eliminazione di tale risorsa. Ad esempio, IAM supporta circa 40 operazioni per una risorsa utente, incluse le seguenti:

- `CreateUser`
- `DeleteUser`
- `GetUser`
- `UpdateUser`

Per consentire a un'entità principale di eseguire un'operazione, devi includere le azioni necessarie in una policy da applicare all'entità principale o alla risorsa interessata. Per visualizzare un elenco di azioni, tipi di risorse e chiavi di condizione supportate da ciascun servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi.

Risorse

Dopo aver AWS approvato le operazioni contenute nella richiesta, queste possono essere eseguite sulle risorse correlate all'interno del tuo account. Una risorsa è un oggetto esistente all'interno di un servizio. Esempi sono un'istanza Amazon EC2, un utente IAM e un bucket Amazon S3. Il servizio definisce un insieme di azioni che possono essere eseguite su ogni risorsa. Se crei una richiesta per eseguire un'azione non correlata su una risorsa, la richiesta viene negata. Ad esempio, se richiedi di eliminare un ruolo IAM ma fornisci una risorsa di gruppo IAM, la richiesta ha esito negativo. Per visualizzare le tabelle dei AWS servizi che identificano le risorse interessate da un'azione, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Panoramica della gestione delle AWS identità: utenti

Puoi concedere l'accesso Account AWS ai tuoi utenti specifici e fornire loro autorizzazioni specifiche per accedere alle risorse del tuo Account AWS. Puoi utilizzare sia IAM che AWS IAM Identity Center

per creare nuovi utenti o federare utenti esistenti in. AWS La differenza principale tra i due è che agli utenti IAM vengono concesse credenziali a lungo termine per le AWS risorse, mentre gli utenti di IAM Identity Center dispongono di credenziali temporanee che vengono stabilite ogni volta che l'utente accede. AWS Come best practice, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché come utenti IAM. Uno degli usi principali per gli utenti IAM consiste nel fornire ai carichi di lavoro che non possono utilizzare i ruoli IAM la possibilità di effettuare richieste programmatiche ai AWS servizi utilizzando l'API o la CLI.

Argomenti

Solo primo accesso: credenziali utente root

Quando ne crei uno Account AWS, inizi con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM. Solo le policy di controllo dei servizi (SCP) nelle organizzazioni possono limitare le autorizzazioni concesse all'utente root.

Utenti IAM e utenti nel Centro identità IAM

Gli utenti IAM non sono account separati, ma utenti all'interno del tuo account. Ciascun utente può disporre della propria password per accedere alla AWS Management Console. Puoi anche assegnare a ciascun utente una diversa chiave di accesso, per consentirgli di apportare richieste programmatiche e utilizzare le risorse del tuo account.

Agli utenti IAM vengono concesse credenziali a lungo termine per AWS le tue risorse. Come best practice, non creare utenti IAM con credenziali a lungo termine per i tuoi utenti. Richiedi invece ai tuoi utenti umani di utilizzare credenziali temporanee per l'accesso. AWS

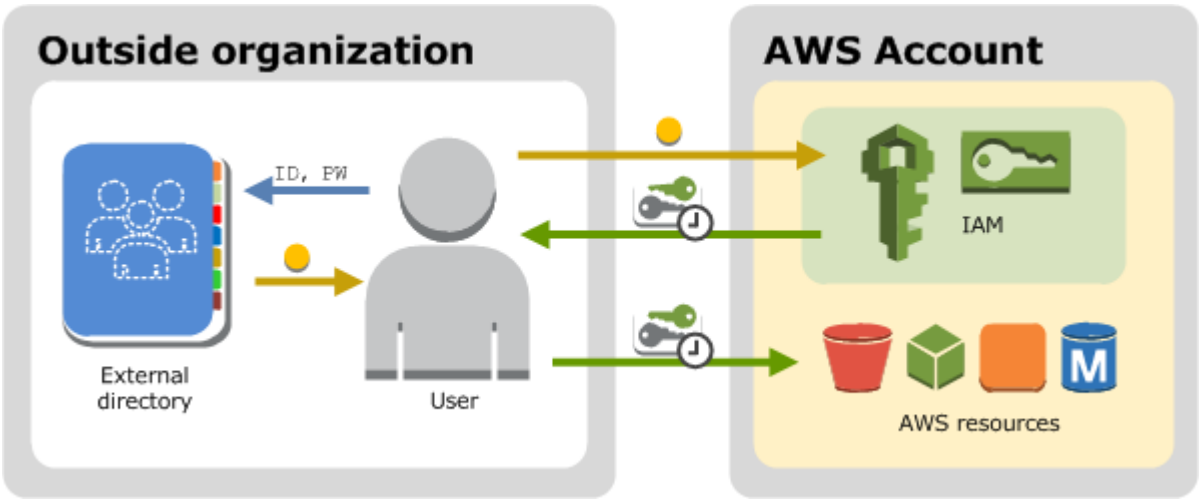
> **Note**
>
> Per gli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le chiavi di accesso all'occorrenza. Per ulteriori informazioni, consulta Aggiornamento delle chiavi di accesso.

Al contrario, utenti in Centro identità AWS IAM vengono concesse credenziali a breve termine per le tue AWS risorse. Per una gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center (IAM Identity Center) per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. IAM Identity Center viene configurato automaticamente con una directory Identity Center come fonte di identità predefinita in cui è possibile creare utenti e gruppi e assegnare il AWS loro livello di accesso alle risorse. Per ulteriori informazioni, consulta Che cos'è AWS IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Federazione di utenti esistenti

Se gli utenti dell'organizzazione dispongono già di una modalità di autenticazione, ad esempio tramite l'accesso alla rete aziendale, non sarà necessario creare utenti IAM o utenti nel Centro identità IAM separati. Puoi invece federare queste identità utente AWS utilizzando IAM o. AWS IAM Identity Center

Il diagramma seguente mostra come un utente può ottenere credenziali di AWS sicurezza temporanee per accedere alle risorse del proprio. Account AWS



La federazione risulta particolarmente utile nei casi seguenti:

- Gli utenti esistono già in una directory aziendale.

Se la directory aziendale è compatibile con Security Assertion Markup Language 2.0 (SAML 2.0), è possibile configurare la directory aziendale per fornire l'accesso Single Sign-On (SSO) ai propri utenti. AWS Management Console Per ulteriori informazioni, consulta Scenari comuni per le credenziali temporanee.

Se la tua directory aziendale non è compatibile con SAML 2.0, puoi creare un'applicazione di identity broker per fornire l'accesso Single Sign-On (SSO) ai tuoi utenti. AWS Management Console Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

Se la directory aziendale è Microsoft Active Directory, è possibile utilizzare AWS IAM Identity Center per connettere una directory autogestita in Active Directory o una directory AWS Directory Serviceper stabilire un rapporto di fiducia tra la directory aziendale e la propria Account AWS.

Se utilizzi un provider di identità (IdP) esterno come Okta o Microsoft Entra per gestire gli utenti, puoi AWS IAM Identity Center utilizzarlo per stabilire un rapporto di fiducia tra il tuo IdP e il tuo. Account AWS Per ulteriori informazioni, consulta Connessione a un provider di identità esterno nella Guida per l'utente di AWS IAM Identity Center .

- I tuoi utenti dispongono di identità Internet.

  Se stai creando un'app mobile o un'applicazione Web che può consentire agli utenti di identificarsi tramite un provider di identità Internet, come Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC), puoi decidere di utilizzare la federazione per accedere ad AWS. Per ulteriori informazioni, consulta Federazione OIDC.

  > **Tip**
  >
  > Per la federazione delle identità con i provider di identità Internet, ti consigliamo di utilizzare Amazon Cognito.

Metodi di controllo degli accessi

Ecco i modi in cui puoi controllare l'accesso alle tue risorse. AWS

Tipo di accesso utente	Perché utilizzarlo?	Dove si possono trovare ulteriori informazioni?
Accesso single sign-on per gli utenti, come gli utenti della forza lavoro, alle risorse AWS tramite il Centro identità IAM	IAM Identity Center offre un luogo centrale che riunisce l'amministrazione degli utenti e il loro accesso alle Account AWS applicazioni cloud. È possibile configurare un archivio di identità all'interno del Centro identità IAM oppure configurare la federazione con un gestore dell'identità digitale (IdP) esistente. Come best practice di sicurezza, si consiglia di concedere agli utenti umani credenziali limitate alle AWS risorse in base alle esigenze. Gli utenti hanno un'esperienza di accesso più semplice e tu mantieni il controllo sul loro accesso alle risorse da un unico sistema. Il Centro identità IAM supporta l'autenticazione a più fattori (MFA) per una maggiore sicurezza degli account.	Per ulteriori informazioni sulla configurazione del Centro identità IAM, consulta Nozioni di base nella Guida per l'utente di AWS IAM Identity Center . Per ulteriori informazioni sull'uso di MFA nel Centro identità IAM, consulta Autenticazione a più fattori (MFA) nella Guida per l'utente di AWS IAM Identity Center .
Accesso federato per gli utenti, come gli utenti della	Supporti IdPs IAM compatibili con OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). Una volta creato un gestore di	Per ulteriori informazioni sulla federazione e sui gestori di identità IAM, consulta Provider di identità e federazione.

Tipo di accesso utente	Perché utilizzarlo?	Dove si possono trovare ulteriori informazioni?
forza lavoro, ai servizi AWS che utilizzano gestori di identità IAM	identità IAM, dovrai creare uno o più ruoli IAM che possano essere assegnati dinamicam ente a un utente federato.	
Accesso tra più account tra Account AWS	Vuoi condividere l'accesso a determinate AWS risorse con gli utenti di altre Account AWS. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, alcuni dei servizi AWS ti consentono di collegare una policy direttame nte a una risorsa (invece di utilizzare un ruolo come proxy). Queste sono denominate policy basate sull'identità.	Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM. Per ulteriori informazioni sui ruoli collegati al servizio, consulta Uso di ruoli collegati ai servizi. Per ulteriori informazioni sui servizi che supportano l'utilizzo di ruoli collegati ai servizi, consulta AWS servizi che funzionano con IAM. Cerca i servizi che hanno Sì nella colonna Ruolo collegato ai servizi. Per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio, seleziona il link associato a Yes (Sì) nella colonna.

Tipo di accesso utente	Perché utilizzarlo?	Dove si possono trovare ulteriori informazioni?
Credenziali a lungo termine per gli utenti IAM designati nel tuo Account AWS	Potresti avere casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM in. AWS Puoi utilizzare IAM per creare questi utenti IAM nel tuo Account AWS e utilizzare IAM per gestirne le autorizzazioni. Alcuni dei casi d'uso sono i seguenti: • Carichi di lavoro che non possono utilizzare ruoli IAM • AWS Client di terze parti che richiedono l'accesso programmatico tramite chiavi di accesso • Credenziali specifiche del servizio per Amazon Keyspaces AWS CodeCommit • AWS IAM Identity Center non è disponibile per il tuo account e non hai un altro provider di identità Come best practice, negli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le	Per informazioni sulla configurazione di un utente IAM, consulta Creare un utente IAM nel tuo Account AWS. Per ulteriori informazioni sulle chiavi di accesso utente IAM, consultaGestione delle chiavi di accesso per gli utenti IAM. Per ulteriori informazioni sulle credenziali specifiche del servizio per AWS CodeCommit Amazon Keyspaces, consulta e. Usare IAM con CodeCommit: credenziali Git, chiavi SSH e AWS chiavi di accesso Utilizzo di IAM con Amazon Keyspaces (per Apache Cassandra)

Tipo di accesso utente	Perché utilizzarlo?	Dove si possono trovare ulteriori informazioni?
	chiavi di accesso all'occorrenza. Per ulteriori informazioni, consulta Aggiornamento delle chiavi di accesso.	

Panoramica della gestione degli accessi: autorizzazioni e policy

La parte di gestione degli accessi di AWS Identity and Access Management (IAM) ti aiuta a definire cosa può fare un'entità principale in un account. Un'entità principale è una persona o un'applicazione che viene autenticata tramite un'entità IAM (utente o ruolo). La gestione degli accessi viene spesso definita come autorizzazione. Puoi gestire l'accesso AWS creando policy e collegandole a identità o risorse IAM (utenti, gruppi di utenti o ruoli). AWS Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale utilizza un'entità IAM (utente o ruolo) per effettuare una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Policy e autorizzazioni in IAM.

Policy e account

Se gestisci un singolo account in AWS, definisci le autorizzazioni all'interno di quell'account utilizzando le politiche. Se gestisci le autorizzazioni di più account, è più difficile gestire le autorizzazioni per i tuoi utenti. È possibile utilizzare i ruoli IAM, le policy basate su risorse o le liste di controllo accessi (ACL) per le autorizzazioni a più account. Tuttavia, se possiedi più account, ti consigliamo invece di utilizzare il AWS Organizations servizio per aiutarti a gestire tali autorizzazioni. Per ulteriori informazioni, consulta Cos'è AWS Organizations? nella Organizations User Guide.

Policy e utenti

Gli utenti IAM sono identità nel servizio. Quando si crea un utente IAM, l'utente non potrà accedere ad alcun elemento nell'account finché non gli viene concessa l'autorizzazione. È possibile fornire autorizzazioni a un utente creando una policy basata su identità collegata all'utente o a un gruppo a cui appartiene l'utente. L'esempio seguente mostra una policy JSON che consente all'utente di

eseguire tutte le operazioni di Amazon DynamoDB (`dynamodb:*`) sulla tabella `Books` nell'account `123456789012` all'interno della regione `us-east-2`.

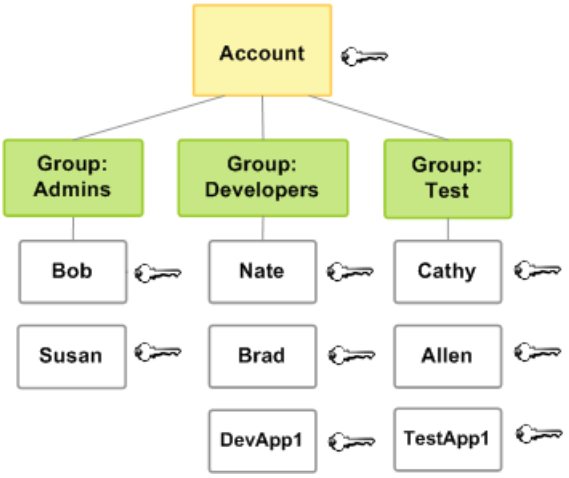
```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "dynamodb:*",
    "Resource": "arn:aws:dynamodb:us-east-2:123456789012:table/Books"
  }
}
```

Dopo il collegamento di questa policy all'utente IAM, l'utente disporrà solo di quelle autorizzazioni DynamoDB. La maggior parte degli utenti ha più policy che insieme rappresentano le autorizzazioni per l'utente.

Le operazioni o le risorse che non sono esplicitamente consentite vengono rifiutate per default. Ad esempio, se la policy precedente è l'unica policy collegata a un utente, quell'utente può solo eseguire operazioni DynamoDB nella tabella `Books`. Le operazioni su tutte le altre tabelle non sono consentite. Allo stesso modo, all'utente non è consentito eseguire alcuna azione in Amazon EC2, Amazon S3 o in qualsiasi altro servizio. AWS Il motivo è che le autorizzazioni per utilizzare questi servizi non sono incluse nella policy.

Policy e gruppi

Puoi organizzare gli utenti IAM in gruppi IAM e collegare una policy a un gruppo. In quel caso, i singoli utenti hanno ancora le proprie credenziali, ma tutti gli utenti in un gruppo dispongono delle autorizzazioni collegate al gruppo. Utilizza i gruppi per facilitare la gestione delle autorizzazioni e per seguire le nostre Best practice per la sicurezza in IAM.



Gli utenti o i gruppi possono avere più policy che sono a loro collegate, le quali concedono diverse autorizzazioni. In questo caso, le autorizzazioni degli utenti sono calcolate in base alla combinazione di policy. Tuttavia, il principio di base è ancora applicato: se l'utente non ha ricevuto un'autorizzazione esplicita per un'operazione e una risorsa, l'utente non dispone di tali autorizzazioni.

Utenti federati e ruoli

Gli utenti federati non hanno identità permanenti come gli utenti IAM. Account AWS Per assegnare le autorizzazioni agli utenti federati, puoi creare un'entità definita come ruolo e definire le autorizzazioni per il ruolo. Quando un utente federato accede AWS, l'utente viene associato al ruolo e gli vengono concesse le autorizzazioni definite nel ruolo. Per ulteriori informazioni, consulta Creazione di un ruolo per un provider di identità di terza parte (federazione).

Policy basate su identità e policy basate su risorse.

Le policy basate su identità sono policy di autorizzazione che si collegano a un'identità IAM, come un utente, un gruppo o un ruolo IAM. Le policy basate su risorse sono policy di autorizzazione che si collegano a una risorsa, come un bucket Amazon S3 o una policy di attendibilità del ruolo IAM.

Le policy basate su identità controllano quali operazioni l'identità può eseguire, su quali risorse e in quali condizioni. Le policy basate su identità possono essere ulteriormente suddivise:

- Politiche gestite: politiche autonome basate sull'identità che puoi allegare a più utenti, gruppi e ruoli nel tuo. Account AWS Puoi utilizzare due tipi di policy gestite:
  - AWS politiche gestite: politiche gestite create e gestite da. AWS Se non conosci l'utilizzo delle politiche, ti consigliamo di iniziare utilizzando le politiche AWS gestite.

- Policy gestite dal cliente: le policy gestite che sono create e gestite nel tuo Account AWS. Le policy gestite dai clienti offrono un controllo più preciso sulle policy rispetto alle policy AWS gestite. Puoi creare, modificare e convalidare una policy IAM nell'editor visivo oppure creando direttamente il documento di policy JSON. Per ulteriori informazioni, consulta Creazione di policy IAM e Modifica delle policy IAM.
- Policy in linea: le policy che sono create, gestite e direttamente incorporate in un singolo utente, gruppo o ruolo. Nella maggior parte dei casi, non è consigliato l'uso di policy inline.

Le policy basate su risorse controllano quali operazioni uno specifico principale può eseguire, su quale risorsa e in quali condizioni. Le policy basate risorse sono policy inline. Non esistono policy gestite basate su risorse. Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse.

Il servizio IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Poiché un ruolo IAM è sia un'identità sia una risorsa che supporta policy basate su risorse, a un ruolo IAM è necessario collegare sia una policy di attendibilità che una policy basata su identità. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Per capire in che modo i ruoli IAM si differenziano da altre policy basate su risorse, consulta Accesso alle risorse multi-account in IAM.

Per scoprire quali servizi supportano le policy basate su risorse, consulta la pagina AWS servizi che funzionano con IAM. Per ulteriori informazioni sulle policy basate su risorse, consulta la pagina Policy basate sulle identità e policy basate su risorse.

A cosa serve ABAC? AWS

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti o ruoli) e alle AWS risorse. È possibile creare una singola policy ABAC o un piccolo insieme di policy per i principali IAM. Queste policy ABAC possono essere definite affinché autorizzino le operazioni quando il tag dell'entità corrisponde al tag della risorsa. La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Ad esempio, è possibile creare tre ruoli associando a essi un tag con la chiave `access-project` e impostando il valore del tag del primo ruolo a `Heart`, del secondo a `Star` e del terzo a `Lightning`. È quindi possibile utilizzare una singola policy che consenta l'accesso quando il ruolo e la risorsa

sono contrassegnati con lo stesso valore del tag `access-project`. Per un tutorial dettagliato che dimostra come utilizzare ABAC in AWS, vedi. Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag Per ulteriori informazioni sui servizi a supporto di ABAC, consulta AWS servizi che funzionano con IAM.

Confronto di ABAC con il modello RBAC tradizionale

Il modello di autorizzazione tradizionale utilizzato in IAM è chiamato controllo dell'accesso basato sul ruolo (Role-Based Access Control, RBAC). RBAC definisce le autorizzazioni in base alle mansioni lavorative di una persona, note al di fuori di AWS come ruolo. All'interno AWS di un ruolo di solito si fa riferimento a un ruolo IAM, che è un'identità in IAM che puoi assumere. IAM include policy gestite per le funzioni processo che allineano le autorizzazioni a una funzione di processo in un modello RBAC.

In IAM, è possibile implementare RBAC creando diverse policy per diverse mansioni lavorative. Quindi è possibile collegare le policy alle identità (utenti IAM, gruppi di utenti o ruoli IAM). Come best practice si suggerisce di concedere le autorizzazioni minime necessarie per la mansione lavorativa. Questo approccio è noto come concessione dei privilegi minimi. Tale risultato si ottiene elencando le risorse specifiche a cui la mansione lavorativa può accedere. Lo svantaggio di utilizzare il modello RBAC tradizionale è che, nel momento in cui i dipendenti aggiungono nuove risorse, per consentire l'accesso a esse è necessario aggiornare le policy.

Ad esempio, si supponga di disporre di tre progetti denominati `Heart`, `Star` e `Lightning`, su cui lavorano i dipendenti. Si crea un ruolo IAM per ogni progetto. È quindi possibile collegare le policy a ciascun ruolo IAM per definire le risorse a cui può accedere chiunque sia autorizzato ad assumere il ruolo. Se un dipendente cambia mansione all'interno dell'azienda, è necessario assegnargli un ruolo IAM differente. Le persone o i programmi possono essere assegnati a più di un ruolo. Tuttavia, il progetto `Star` potrebbe richiedere risorse aggiuntive, ad esempio un nuovo container Amazon EC2. In tal caso, è necessario aggiornare la policy collegata al ruolo `Star` per specificare la nuova risorsa del container. In caso contrario, i membri del progetto `Star` non potranno accedere al nuovo container.

Il modello ABAC offre i seguenti vantaggi rispetto al modello RBAC tradizionale:

- Le autorizzazioni ABAC si ridimensionano con l'innovazione. Non è più necessario che un amministratore aggiorni le policy esistenti per consentire l'accesso a nuove risorse. Ad esempio, si supponga di aver progettato la strategia ABAC con il tag `access-project`. Uno sviluppatore utilizza il ruolo con il tag `access-project` = `Heart`. Quando le persone del progetto `Heart`

hanno bisogno di risorse Amazon EC2 aggiuntive, lo sviluppatore può creare nuove istanze Amazon EC2 con il tag `access-project` = `Heart`. In questo modo chiunque partecipi al progetto `Heart` può avviare e arrestare tali istanze perché i rispettivi valori di tag corrispondono.

- ABAC richiede meno policy. Poiché non è necessario creare policy diverse per diverse mansioni lavorative, è necessario creare meno policy. Tali policy sono più facili da gestire.
- Utilizzando ABAC, i team possono cambiare e crescere rapidamente. Questo perché le autorizzazioni per le nuove risorse vengono concesse automaticamente in base agli attributi. Ad esempio, se la propria azienda supporta già i progetti `Heart` e `Star` utilizzando ABAC, è facile aggiungere un nuovo progetto `Lightning`. Un amministratore IAM crea un nuovo ruolo con il tag `access-project` = `Lightning`. Non è necessario modificare la policy per supportare un nuovo progetto. Chiunque disponga delle autorizzazioni per assumere il ruolo può creare e visualizzare istanze a cui è stato assegnato il tag `access-project` = `Lightning`. Inoltre, un membro del team potrebbe passare dal progetto `Heart` al progetto `Lightning`. L'amministratore IAM assegna l'utente a un ruolo IAM diverso. Non è necessario modificare le policy di autorizzazione.
- Utilizzando la strategia ABAC e possibile definire autorizzazioni con un maggior livello di granularità. Quando si creano le policy, è consigliabile concedere i privilegi minimi. Utilizzando l'approccio RBAC tradizionale, è necessario scrivere una policy che consenta l'accesso solo a specifiche risorse. Tuttavia, quando si utilizza ABAC, è possibile consentire operazioni su tutte le risorse, ma solo se il tag della risorsa corrisponde al tag dell'entità.
- Con ABAC è possibile utilizzare gli attributi dei dipendenti memorizzati nella directory aziendale. Puoi configurare il tuo provider SAML o OIDC a cui passare i tag di sessione. AWS Quando i tuoi dipendenti si uniscono in AWS, i loro attributi vengono applicati al responsabile risultante in. AWSÈ quindi possibile utilizzare ABAC per consentire o negare le autorizzazioni sulla base di tali attributi.

Per un tutorial dettagliato che dimostra come utilizzare ABAC in AWS, vedi. Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag

Funzioni di sicurezza al di fuori di IAM

Utilizzi IAM per controllare l'accesso alle attività eseguite utilizzando gli strumenti della AWS Management Console riga di AWS comando o le operazioni dell'API di servizio utilizzando gli AWS SDK. Alcuni AWS prodotti offrono anche altri modi per proteggere le proprie risorse. In via esemplificativa, di seguito sono elencati alcuni esempi.

Amazon EC2

In Amazon Elastic Compute Cloud si accede a un'istanza con una coppia di chiavi (per le istanze di Linux) o utilizzando un nome utente e una password (per le istanze di Microsoft Windows).

Per ulteriori informazioni, consulta la seguente documentazione :

- Guida introduttiva alle istanze Amazon EC2 Linux nella Guida per l'utente di Amazon EC2
- Guida introduttiva alle istanze Windows di Amazon EC2 nella Guida per l'utente di Amazon EC2

Amazon RDS

Per accedere al motore di database in Amazon Relational Database Service è necessario utilizzare nome utente e password associati al database.

Per ulteriori informazioni, consulta Nozioni di base su Amazon RDS nella Guida per l'utente di Amazon RDS.

Amazon EC2 e Amazon RDS

In Amazon EC2 e Amazon RDS si utilizzano gruppi di sicurezza per controllare il traffico verso un'istanza o un database.

Per ulteriori informazioni, consulta la seguente documentazione :

- Gruppi di sicurezza di Amazon EC2 per istanze Linux nella Guida per l'utente di Amazon EC2
- Gruppi di sicurezza di Amazon EC2 per istanze Windows nella Guida per l'utente di Amazon EC2
- Gruppi di sicurezza di Amazon RDS nella Guida per l'utente di Amazon RDS

WorkSpaces

In Amazon WorkSpaces, gli utenti accedono a un desktop con un nome utente e una password.

Per ulteriori informazioni, consulta la sezione Getting Started with WorkSpaces nella Amazon WorkSpaces Administration Guide.

Amazon WorkDocs

In Amazon WorkDocs, gli utenti possono accedere ai documenti condivisi accedendo con un nome utente e una password.

Per ulteriori informazioni, consulta Getting Started with Amazon WorkDocs nella Amazon WorkDocs Administration Guide.

Questi metodi di controllo degli accessi non sono parte di IAM. IAM ti consente di controllare il modo in cui questi AWS prodotti vengono amministrati, creando o terminando un'istanza Amazon EC2, configurando nuovi WorkSpaces desktop e così via. Ovvero, IAM consente di controllare i processi eseguiti tramite l'esecuzione di richieste ad Amazon Web Services nonché l'accesso alla AWS Management Console. Tuttavia, IAM non ti aiuta a gestire la sicurezza per attività come l'accesso a un sistema operativo (Amazon EC2), database (Amazon RDS), desktop (Amazon WorkSpaces) o sito di collaborazione (Amazon). WorkDocs

Quando lavori con un AWS prodotto specifico, assicurati di leggere la documentazione per conoscere le opzioni di sicurezza per tutte le risorse che appartengono a quel prodotto.

Collegamenti rapidi per le attività comuni

Utilizza i link seguenti per ottenere maggiori informazioni sui processi comuni associati a IAM.

Accedi per diversi tipi di utente

Accedi alla console IAM selezionando l'utente IAM e inserendo il tuo Account AWS ID o l'alias dell'account. Nella pagina successiva, inserisci il nome utente IAM e la password.

Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per assistenza nell'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Accedi AWS Management Consolecome proprietario dell'account selezionando Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Consulta Che cos'è Accedi ad AWS nella Guida per l'utente di Accedi ad AWS per aiutarti a determinare il tipo di utente e la pagina di accesso.

Gestione delle password per gli utenti

È necessaria una password per accedere a AWS Management Console, incluso l'accesso alle informazioni di fatturazione.

Per il tuo Utente root dell'account AWS, consulta Modifica della password Utente root dell'account AWS nella Guida di AWS Account Management riferimento

Per un utente IAM, consulta Gestione delle password per gli utenti IAM.

Gestione delle autorizzazioni per gli utenti

Utilizzi le policy per concedere le autorizzazioni agli utenti IAM del tuo Account AWS. Gli utenti IAM non dispongono di autorizzazioni al momento della creazione, quindi è necessario aggiungere autorizzazioni per consentire loro di utilizzare le risorse. AWS

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in: AWS IAM Identity Center

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Creating a role for a third-party identity provider (federation) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina Creating a role for an IAM user (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente di IAM.

Per ulteriori informazioni, consulta Gestione di policy IAM.

Elenca gli utenti del tuo Account AWS account e ottieni informazioni sulle loro credenziali

Per informazioni, consulta Recupero dei report delle credenziali per l' Account AWS.

Aggiungere un'autenticazione a più fattori (MFA)

Per aggiungere un dispositivo MFA virtuale, consulta uno dei seguenti argomenti:

- Abilitazione di un dispositivo MFA virtuale per l' Utente root dell'account AWS (console)
- Abilitazione di un dispositivo MFA virtuale per un utente IAM (console)

Per aggiungere una chiave di sicurezza FIDO, consulta uno dei seguenti argomenti:

- Abilita una passkey o una chiave di sicurezza per Utente root dell'account AWS (console)
- Abilita una passkey o una chiave di sicurezza per un altro utente IAM (console)

Per aggiungere un dispositivo MFA hardware, consulta uno dei seguenti argomenti:

- Abilita un token TOTP hardware per Utente root dell'account AWS (console).
- Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)

Ottenere una chiave di accesso

Puoi utilizzare una chiave di accesso per effettuare AWS richieste utilizzando gli AWS SDK, gli strumenti della AWS riga di comando o le operazioni API.

> **Important**
>
> Come best practice, utilizza credenziali di sicurezza temporanee (come i ruoli IAM) invece di creare credenziali a lungo termine come le chiavi di accesso. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Per assistenza per la protezione delle tue chiavi di accesso, consulta Protezione delle chiavi di accesso.

Per informazioni sulla gestione delle chiavi di accesso per un utente IAM, consulta Gestione delle chiavi di accesso per gli utenti IAM.

Per ulteriori informazioni sulle credenziali di sicurezza disponibili per te Account AWS, consulta Credenziali AWS di sicurezza.

Aggiunta di tag alle risorse IAM

Puoi contrassegnare con i tag le seguenti risorse IAM:

- Utenti IAM
- Ruoli IAM
- Policy gestite dal cliente
- Provider di identità
- Certificati server
- Dispositivi MFA virtuali

Per ulteriori informazioni sui tag in IAM, consulta Tagging delle risorse IAM.

Per ulteriori informazioni sull'utilizzo dei tag per controllare l'accesso alle AWS risorse, consulta. Controllo dell'accesso alle AWS risorse tramite tag

Visualizzare operazioni, risorse e chiavi di condizione per tutti i servizi

Questa documentazione di riferimento può aiutare a scrivere policy IAM dettagliate. Ogni servizio AWS definisce le operazioni, le risorse e le chiavi di condizione contestuali utilizzate nelle policy IAM. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Inizia con tutti AWS

Questa serie di documenti è principalmente dedicata al servizio IAM. Per informazioni su come iniziare AWS e utilizzare più servizi per risolvere un problema, ad esempio la creazione e il lancio del primo progetto, consulta il Centro risorse per iniziare.

Ricerca con la console IAM

Per trovare rapidamente le risorse IAM, puoi utilizzare la pagina di ricerca della console IAM. Puoi utilizzare la ricerca su console per individuare le chiavi di accesso relative al tuo account, alle entità IAM (come utenti, gruppi, ruoli, provider di identità), alle politiche per nome e altro ancora.

La funzione di ricerca della console IAM è in grado di trovare quanto segue:

- Nomi di entità IAM che corrispondono alle parole chiave della ricerca (per utenti, gruppi, ruoli, provider di identità e policy)
- Attività corrispondenti alle parole chiave usate per la ricerca

La funzionalità di ricerca della console IAM non restituisce informazioni su IAM Access Analyzer.

Ogni riga visualizzata nei risultati della ricerca è un link attivo. Ad esempio, puoi selezionare il nome utente nei risultati di ricerca, per andare direttamente alla pagina dei dettagli dell'utente. In alternativa, puoi selezionare collegamento all'operazione, come ad esempio Create User (Crea utente), per accedere alla pagina Create User (Crea utente).

> **Note**
>
> Per le ricerche delle chiavi di accesso è necessario immettere nella casella di ricerca l'ID completo della chiave di accesso. Il risultato della ricerca mostra l'utente associato a tale

chiave. A questo punto, potrai andare direttamente alla pagina dell'utente e gestirne la chiave di accesso.

Utilizzo della funzione di ricerca della console IAM

Utilizza la pagina Cerca della console IAM per trovare gli elementi relativi a un determinato account.

Come cercare gli elementi nella console IAM

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console, seleziona il servizio IAM.
3. Nel riquadro di navigazione selezionare Search (Cerca).
4. Nella casella Search (Cerca) digitare le parole chiave usate per la ricerca.
5. Nell'elenco dei risultati della ricerca selezionare un link per passare alla parte corrispondente della console.

Icone dei risultati della ricerca nella console IAM

Le icone riportate di seguito consentono di identificare i tipi di elementi trovati mediante una ricerca:

Icon	Descrizione
	Utenti IAM
	Gruppi IAM
	Ruoli IAM
	Policy IAM
	Attività come la creazione di utenti o il collegamento di policy

Icon	Descrizione
	Risultati della parola chiave **delete**

Esempi di frasi da cercare

Per le ricerche in IAM puoi utilizzare le frasi riportate di seguito: Sostituisci i termini in corsivo con i nomi di veri utenti IAM, gruppi, ruoli, chiavi di accesso, policy o gestori di identità da individuare.

- ***user_name*** o ***group_name*** o ***role_name*** o ***policy_name*** o ***identity_provider_name***
- ***access_key***
- **add user *user_name* to groups** o **add users to group *group_name***
- **remove user *user_name* from groups**
- **delete *user_name*** o **delete *group_name*** o **delete *role_name*** o **delete *policy_name*** o **delete *identity_provider_name***
- **manage access keys *user_name***
- **manage signing certificates *user_name***
- **users**
- **manage MFA for *user_name***
- **manage password for *user_name***
- **create role**
- **password policy**
- **edit trust policy for role *role_name***
- **show policy document for role *role_name***
- **attach policy to *role_name***
- **create managed policy**
- **create user**
- **create group**
- **attach policy to *group_name***
- **attach entities to *policy_name***
- **detach entities from *policy_name***

Creazione di AWS Identity and Access Management risorse con AWS CloudFormation

AWS Identity and Access Management è integrato con AWS CloudFormation, un servizio che consente di modellare e configurare le AWS risorse in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Crei un modello che descrive tutte le AWS risorse che desideri (come chiavi di accesso, gruppi, politiche di gruppo, profili di istanza, politiche gestite, provider OIDC, politiche in linea, ruoli, politiche di ruolo, provider SAML, certificati server, ruoli collegati ai servizi, utenti (e aggiunta di utenti ai gruppi), politiche utente e dispositivi MFA virtuali) e fornisce e AWS CloudFormation configura tali risorse per te.

Quando lo utilizzi AWS CloudFormation, puoi riutilizzare il modello per configurare le risorse IAM in modo coerente e ripetuto. Descrivi le tue risorse una sola volta, quindi fornisci le stesse risorse più e più volte in più Account AWS regioni.

IAM e AWS CloudFormation modelli

Per fornire e configurare le risorse per IAM e i servizi correlati, è necessario conoscere AWS CloudFormation i modelli. I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri fornire nei tuoi AWS CloudFormation stack. Se non conosci JSON o YAML, puoi usare AWS CloudFormation Designer per iniziare a usare i modelli. AWS CloudFormation Per ulteriori informazioni, consulta Che cos'è AWS CloudFormation Designer? nella Guida per l'utente di AWS CloudFormation .

IAM supporta la creazione di chiavi di accesso, gruppi, policy di gruppo, profili di istanza, policy gestite, provider OIDC, policy in linea, ruoli, policy di ruolo, provider SAML, certificati server, ruoli collegati ai servizi, utenti (e aggiunta di utenti ai gruppi), policy utente e dispositivi MFA virtuali in. AWS CloudFormationPer ulteriori informazioni, inclusi esempi di modelli JSON e YAML per risorse IAM, consulta il riferimento al tipo di risorsa nella Guida per l'utente.AWS Identity and Access Management AWS CloudFormation

Puoi anche creare modelli che creano risorse correlate, come ruoli e politiche gestite.

Scopri di più su AWS CloudFormation

Per ulteriori informazioni AWS CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation

- AWS CloudFormation Guida per l'utente
- AWS CloudFormation Documentazione di riferimento delle API
- AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando

Utilizzo AWS CloudShell per l'utilizzo con AWS Identity and Access Management

AWS CloudShell è una shell preautenticata basata su browser che puoi avviare direttamente da. AWS Management ConsoleÈ possibile eseguire AWS CLI comandi sui AWS servizi (incluso AWS Identity and Access Management) utilizzando la shell preferita (Bash PowerShell o Z shell). E puoi farlo senza dover scaricare o installare strumenti da riga di comando.

Si avvia AWS CloudShell da AWS Management Console, e AWS le credenziali utilizzate per accedere alla console sono automaticamente disponibili in una nuova sessione di shell. Questa preautenticazione degli AWS CloudShell utenti consente di ignorare la configurazione delle credenziali quando si interagisce con AWS servizi come IAM utilizzando la AWS CLI versione 2 (preinstallata nell'ambiente di calcolo della shell).

Ottenere le autorizzazioni IAM per AWS CloudShell

Utilizzando le risorse di gestione degli accessi fornite da AWS Identity and Access Management, gli amministratori possono concedere le autorizzazioni agli utenti IAM in modo che possano accedere AWS CloudShell e utilizzare le funzionalità dell'ambiente.

Il modo più rapido per un amministratore di concedere l'accesso agli utenti è tramite una AWS policy gestita. Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. La seguente policy AWS gestita per CloudShell può essere allegata alle identità IAM:

- `AWSCloudShellFullAccess`: concede l'autorizzazione all'uso AWS CloudShell con accesso completo a tutte le funzionalità.

Se desideri limitare l'ambito di azioni che un utente IAM può eseguire AWS CloudShell, puoi creare una policy personalizzata che utilizzi la policy `AWSCloudShellFullAccess` gestita come modello. Per ulteriori informazioni sulla limitazione delle azioni disponibili per gli utenti in CloudShell, consulta Gestire AWS CloudShell l'accesso e l'utilizzo con le politiche IAM nella Guida per l'AWS CloudShell utente.

Interagire con IAM utilizzando AWS CloudShell

Dopo l'avvio AWS CloudShell da AWS Management Console, puoi iniziare immediatamente a interagire con IAM utilizzando l'interfaccia a riga di comando.

> **Note**
>
> Quando si utilizza AWS CLI in AWS CloudShell, non è necessario scaricare o installare risorse aggiuntive. Inoltre, poiché hai già eseguito l'autenticazione alla shell, non è necessario configurare le credenziali prima di effettuare chiamate.

Crea un gruppo IAM e aggiungi un utente IAM al gruppo utilizzando AWS CloudShell

L'esempio seguente utilizza la creazione CloudShell di un gruppo IAM, l'aggiunta di un utente IAM al gruppo e la verifica dell'esito positivo del comando.

1. Da AWS Management Console, puoi avviarlo CloudShell scegliendo le seguenti opzioni disponibili nella barra di navigazione:

   - Scegli l' CloudShell icona.
   - Inizia a digitare «cloudshell» nella casella di ricerca, quindi scegli l'opzione. CloudShell

2. Per creare un gruppo IAM, inserisci il seguente comando nella riga di comando. CloudShell In questo esempio abbiamo denominato il gruppo east_coast:

   ```
   aws iam create-group --group-name east_coast
   ```

   Se la chiamata ha esito positivo, la riga di comando visualizza una risposta del servizio simile al seguente output:

   ```
   {
      "Group": {
          "Path": "/",
          "GroupName": "east_coast",
          "GroupId": "AGPAYBDBW4JBY3EXAMPLE",
          "Arn": "arn:aws:iam::111122223333:group/east_coast",
          "CreateDate": "2023-09-11T21:02:21+00:00"
       }
   ```

```
        }
```

3. Per aggiungere un utente al gruppo creato, utilizza il seguente comando, specificando il nome e il nome utente del gruppo. In questo esempio abbiamo denominato il gruppo east_coast e l'utente johndoe:

```
aws iam add-user-to-group --group-name east_coast --user-name johndoe
```

4. Per verificare che l'utente sia nel gruppo, utilizza il seguente comando, specificando il nome del gruppo. In questo esempio continuiamo a utilizzare il gruppo east_coast:

```
aws iam get-group --group-name east_coast
```

   Se la chiamata ha esito positivo, la riga di comando visualizza una risposta del servizio simile al seguente output:

```
      {
        "Users": [
          {
              "Path": "/",
              "UserName": "johndoe",
              "UserId": "AIDAYBDBW4JBXGEXAMPLE",
              "Arn": "arn:aws:iam::552108220995:user/johndoe",
              "CreateDate": "2023-09-11T20:43:14+00:00",
              "PasswordLastUsed": "2023-09-11T20:59:14+00:00"
          }
        ],
        "Group": {
              "Path": "/",
              "GroupName": "east_coast",
              "GroupId": "AGPAYBDBW4JBY3EXAMPLE",
              "Arn": "arn:aws:iam::111122223333:group/east_coast",
              "CreateDate": "2023-09-11T21:02:21+00:00"
           }
      }
```

Utilizzo di IAM con un AWS SDK

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici relativi a IAM, consulta la sezione Esempi di codice per IAM che utilizza gli AWS SDK.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Configurazione di IAM

> **Important**
>
> Le migliori pratiche IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché utilizzare utenti IAM con credenziali a lungo termine.

AWS Identity and Access Management (IAM) ti aiuta a controllare in modo sicuro l'accesso ad Amazon Web Services (AWS) e alle risorse del tuo account. IAM può anche mantenere le credenziali di accesso private. Non è necessario registrarsi specificamente per utilizzare IAM. L'uso di IAM non comporta alcun costo.

Utilizza IAM per concedere a identità, ad esempio ruoli e ruoli, l'accesso alle risorse nell'account. Ad esempio, puoi utilizzare IAM con gli utenti esistenti nella tua directory aziendale che gestisci esternamente AWS oppure puoi creare utenti da AWS utilizzare AWS IAM Identity Center. Le identità federate assumono ruoli IAM definiti per accedere alle risorse di cui hanno bisogno. Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

> **Note**
>
> IAM è integrato con diversi AWS prodotti. Per un elenco di servizi che supportano IAM, consulta AWS servizi che funzionano con IAM.

Argomenti

Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Preparazione per le autorizzazioni con privilegi minimi

L'utilizzo delle autorizzazioni con privilegi minimi è uno dei suggerimenti di best practice di IAM. Lo scopo delle autorizzazioni con privilegi minimi è concedere agli utenti solo le autorizzazioni richieste per eseguire una determinata attività. Durante la configurazione, considera come intendi supportare le autorizzazioni con privilegi minimi. Sia l'utente root sia l'utente amministratore dispongono di autorizzazioni avanzate che non sono necessarie per le attività quotidiane. Mentre stai imparando

a conoscere AWS e testare diversi servizi, ti consigliamo di creare almeno un utente aggiuntivo in IAM Identity Center con autorizzazioni inferiori da utilizzare in diversi scenari. È possibile utilizzare le policy IAM per definire le operazioni che possono essere eseguite su risorse specifiche in determinate condizioni e connettersi quindi a tali risorse con l'account con meno privilegi.

Se utilizzi il Centro identità IAM, per iniziare considera l'uso dei set di autorizzazioni del Centro identità IAM stesso. Per ulteriori informazioni, consulta la pagina Creazione di un set di autorizzazioni nella Guida per l'utente di Centro identità IAM.

Se non utilizzi Centro identità IAM, utilizza i ruoli IAM per definire le autorizzazioni per diverse entità IAM. Per ulteriori informazioni, consulta Creazione di ruoli IAM.

Sia i ruoli IAM che i set di autorizzazioni IAM Identity Center possono utilizzare policy AWS gestite basate sulle funzioni lavorative. Per i dettagli sulle autorizzazioni concesse da queste policy, consulta AWS politiche gestite per le funzioni lavorative.

> **Important**
>
> Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici, poiché sono disponibili per l'uso da parte di tutti i clienti. AWS Dopo la configurazione, consigliamo di utilizzare il Sistema di analisi degli accessi IAM per generare policy con privilegi minimi in funzione dell'attività di accesso collegata in AWS CloudTrail. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Metodi di gestione IAM

Puoi gestire IAM utilizzando la AWS console, l'interfaccia a AWS riga di comando o tramite le interfacce applicative (API) negli SDK associati. Durante la configurazione, valuta quali metodi desideri supportare e in che modo intendi supportare i diversi utenti.

Argomenti

- AWS Console
- AWS Interfaccia a riga di comando (CLI) e kit di sviluppo software (SDK)

AWS Console

La console di AWS gestione è un'applicazione Web che comprende e fa riferimento a un'ampia raccolta di console di servizio per la gestione AWS delle risorse. Quando effettui l'accesso per la prima volta, visualizzi la home page della console. La home page fornisce l'accesso a ciascuna console di servizio e offre un'unica posizione per accedere alle informazioni per l'esecuzione delle attività AWS correlate. I servizi e le applicazioni disponibili dopo l'accesso alla console dipendono dalle AWS risorse a cui si è autorizzati ad accedere. È possibile ottenere le autorizzazioni per le risorse assumendo un ruolo, facendo parte di un gruppo al quale sono state concesse le autorizzazioni oppure ricevendo un'autorizzazione esplicita. Per un AWS account autonomo, l'utente root o l'amministratore IAM configura l'accesso alle risorse. Infatti AWS Organizations, l'account di gestione o l'amministratore delegato configura l'accesso alle risorse.

Se prevedi che persone utilizzino la console di AWS gestione per gestire AWS le risorse, ti consigliamo di configurare gli utenti con credenziali temporanee come best practice di sicurezza. Gli utenti IAM che hanno assunto un ruolo, gli utenti federati e gli utenti in Centro identità IAM dispongono di credenziali temporanee, mentre l'utente IAM e l'utente root dispongono di credenziali a lungo termine. Le credenziali dell'utente root forniscono l'accesso completo all' Account AWS, mentre gli altri utenti dispongono di credenziali che forniscono l'accesso alle risorse loro consentite dalle policy IAM.

L'esperienza di accesso è diversa per i diversi tipi di utenti. AWS Management Console

- Gli utenti IAM e l'utente root accedono dall'URL di AWS accesso principale (https://signin.aws.amazon.com). Una volta effettuato l'accesso, hanno accesso alle risorse dell'account per il quale hanno ricevuto l'autorizzazione.

  Per accedere come utente root è necessario disporre dell'indirizzo e-mail e della password dell'utente root.

  Per accedere come utente IAM devi disporre del Account AWS numero o dell'alias, del nome utente IAM e della password utente IAM.

  Ti consigliamo di limitare gli utenti IAM del tuo account a situazioni specifiche che richiedono credenziali a lungo termine, ad esempio per l'accesso di emergenza, e di utilizzare l'utente root solo per le attività che richiedono le credenziali dell'utente root.

  Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. La volta successiva che l'utente accede a qualsiasi pagina di

AWS Management Console, la console utilizza il cookie per reindirizzare l'utente alla pagina di accesso dell'account.

Per evitare che le tue credenziali vengano riutilizzate dopo il tuo accesso, esci dalla console al termine della sessione.

- Gli utenti di IAM Identity Center accedono utilizzando un portale di AWS accesso specifico, unico per la loro organizzazione. Una volta effettuato l'accesso, possono scegliere a quale account o applicazione accedere. Se scelgono di accedere a un account, scelgono quale set di autorizzazioni utilizzare per la sessione di gestione.
- Gli utenti federati gestiti in un provider di identità esterno collegato a un Account AWS eseguono l'accesso tramite un portale di accesso aziendale personalizzato. Le AWS risorse disponibili per gli utenti federati dipendono dalle politiche selezionate dalla loro organizzazione.

> **Note**
>
> Per fornire un ulteriore livello di sicurezza, l'utente root, gli utenti IAM e gli utenti di IAM Identity Center possono far verificare l'autenticazione a più fattori (MFA) prima AWS di concedere l'accesso alle risorse. AWS Quando l'MFA è abilitata, devi avere accesso anche al dispositivo MFA per accedere.

Per ulteriori informazioni su come diversi utenti accedono alla console di gestione, consulta Accedere alla console di AWS gestione nella Guida per l'utente di accesso.AWS

AWS Interfaccia a riga di comando (CLI) e kit di sviluppo software (SDK)

Gli utenti di Centro identità IAM e IAM utilizzano metodi diversi per autenticare le proprie credenziali quando si autenticano tramite la CLI o le interfacce applicative (API) negli SDK associati.

Le credenziali e le impostazioni di configurazione si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Alcune posizioni hanno la precedenza su altre.

Sia Centro identità IAM sia IAM forniscono chiavi di accesso che possono essere utilizzate con la CLI o l'SDK. Le chiavi di accesso Centro identità IAM sono credenziali temporanee che possono essere aggiornate automaticamente e sono consigliate rispetto alle chiavi di accesso a lungo termine associate agli utenti IAM.

Puoi gestire l' Account AWS utilizzo della CLI o dell'SDK AWS CloudShell dal tuo browser. Se utilizzi CloudShell per eseguire comandi CLI o SDK, devi prima accedere alla console. Le autorizzazioni per l'accesso alle AWS risorse si basano sulle credenziali utilizzate per accedere alla console. A seconda della tua esperienza, potresti ritenere che la CLI sia un metodo più efficiente per gestire il tuo Account AWS.

Per lo sviluppo di applicazioni, puoi scaricare la CLI o l'SDK sul tuo computer e accedere dal prompt dei comandi o da una finestra Docker. In questo scenario, configuri l'autenticazione e le credenziali di accesso come parte dello script della CLI o dell'applicazione SDK. È possibile configurare l'accesso a livello di programmazione alle risorse in diversi modi, a seconda dell'ambiente e dell'accesso a disposizione.

- Le opzioni consigliate per l'autenticazione del codice locale con il AWS servizio sono IAM Identity Center e IAM Roles Anywhere
- Le opzioni consigliate per l'autenticazione del codice in esecuzione all'interno di un AWS ambiente consistono nell'utilizzare i ruoli IAM o le credenziali IAM Identity Center.

Se utilizzi Centro identità IAM, puoi ottenere credenziali a breve termine dalla pagina iniziale del portale di accesso ad AWS in cui scegli il tuo set di autorizzazioni. Queste credenziali hanno una durata definita e non si aggiornano automaticamente. Se desideri utilizzare queste credenziali, dopo aver effettuato l'accesso al AWS portale, scegli Account AWS e quindi scegli il set di autorizzazioni. Seleziona Accesso da riga di comando o accesso programmatico per visualizzare le opzioni che puoi utilizzare per accedere alle AWS risorse a livello di codice o dalla CLI. Per ulteriori informazioni su questi metodi, consulta la pagina Ottenimento e aggiornamento di credenziali temporanee nella Guida per l'utente di Centro identità IAM. Queste credenziali vengono spesso utilizzate durante lo sviluppo di applicazioni per testare rapidamente il codice.

Ti consigliamo di utilizzare le credenziali IAM Identity Center che si aggiornano automaticamente durante l'automazione dell'accesso alle risorse. AWS Se hai configurato utenti e set di autorizzazioni in Centro identità IAM, utilizza il comando `aws configure sso` per impiegare una procedura guidata da linea di comando che ti aiuterà a identificare le credenziali a tua disposizione e a memorizzarle in un profilo. Per ulteriori informazioni sulla configurazione del profilo, consulta la pagina Configurazione del profilo con la procedura guidata `aws configure sso` della Guida per l'utente dell'interfaccia della linea di comando AWS per la versione 2.

> **Note**
>
> Molte applicazioni di esempio utilizzano chiavi di accesso a lungo termine associate agli utenti IAM o all'utente root. È consigliabile utilizzare le credenziali a lungo termine solo all'interno di un ambiente di sperimentazione (sandbox) come parte di un'esercitazione. Esamina le alternative alle chiavi di accesso a lungo termine e pianifica la transizione del codice per utilizzare credenziali alternative, come le credenziali Centro identità IAM o i ruoli IAM, il prima possibile. Dopo la transizione del codice, elimina le chiavi di accesso.

Per ulteriori informazioni sulla configurazione della CLI, consulta Installare o aggiornare la versione più recente della AWS CLI nella Guida per l'utente dell'interfaccia a riga di comando per AWS la versione 2 e Credenziali di autenticazione e accesso nella Guida per l'utente dell'interfaccia a AWS riga di comando

Per ulteriori informazioni sulla configurazione dell'SDK, consulta le pagine Autenticazione a Centro identità IAM nella Guida di riferimento agli SDK e agli strumenti AWS e IAM Roles Anywhere nella Guida di riferimento agli SDK e agli strumenti AWS .

Il tuo Account AWS ID e il suo alias

Gli utenti IAM dell'account accedono utilizzando un URL Web che include l'alias dell'account o un ID dell'account. Se non disponi dell'URL, la pagina di AWS accesso richiede che tu fornisca l'alias o l'ID dell' Account AWS account.

Se non conosci l'ID o l'alias del tuo account:

- Controlla la cronologia del tuo browser. Se hai effettuato l'accesso in precedenza, potrebbe essere archiviato nei tuoi siti Web recenti.
- Se hai configurato la AWS CLI o un AWS SDK con le credenziali del tuo account, puoi ottenere l'ID dell'account dai tuoi file di configurazione.
- Chiedi all'amministratore locale o al proprietario dell'account, AWS non è possibile fornire gli ID dell'account agli utenti.

> **Tip**
>
> Per creare un segnalibro per la pagina di accesso al tuo account nel browser Web, devi digitare manualmente l'URL di accesso come voce del segnalibro. Non utilizzare la funzionalità "aggiungi questa pagina ai preferiti" del tuo browser Web perché acquisisce informazioni specifiche della sessione corrente del browser che interferiscono con le visite future alla pagina di accesso.

Argomenti

Visualizza il tuo Account AWS ID

Puoi visualizzare l'ID del tuo account Account AWS utilizzando i seguenti metodi.

Visualizzazione dell'ID account tramite la console

L'ID dell'account viene visualizzato nella dashboard IAM nella Account AWS sezione. Esistono altri modi per visualizzare l'ID dell'account nella console a seconda del tipo di utente. Se hai assunto un ruolo, le credenziali di sicurezza non sono disponibili.

Tipo di utente	Procedura
Utente root	Nella barra di navigazione in alto a destra, scegli il tuo nome utente, quindi scegli Credenziali di sicurezza. Il numero dell'account viene visualizzato sotto Identificatori dell'acco unt.
Utente IAM	Nella barra di navigazione in alto a destra, scegli il tuo nome utente, l'ID dell'account viene visualizzato sopra il tuo nome utente. Scegli Security Credentials (Credenziali di sicurezza).

Tipo di utente	Procedura
	Il numero dell'account viene visualizzato sotto Dettagli dell'account.
Utente federato	Nella barra di navigazione in alto a destra, scegli il tuo nome utente, l'ID dell'account viene visualizzato sopra il tuo nome utente.
Ruolo presunto	Nella barra di navigazione in alto a destra, scegli l'icona Supporto, quindi scegli Support Center dall'elenco. Il numero di account a 12 cifre (ID) correntemente collegato viene visualizzato nel pannello di navigazione Centro assistenza.

Visualizza l'ID del tuo account utilizzando il AWS CLI

Utilizza il comando seguente per visualizzare l'ID utente, l'ID account e l'ARN utente:

- aws sts get-caller-identity

Visualizzazione dell'ID tramite l'API

Utilizza la seguente API per visualizzare l'ID utente, l'ID account e l'ARN utente:

- GetCallerIdentity

Informazioni sugli alias degli account

Se desideri che l'URL della tua pagina di accesso contenga il nome della tua azienda (o un altro identificativo descrittivo) anziché il tuo Account AWS ID, puoi creare un alias per l'account. Questa sezione fornisce informazioni sugli Account AWS alias ed elenca le operazioni API utilizzate per creare un alias.

Per impostazione predefinita, l'URL della pagina di accesso ha il formato seguente.

```
https://Your_Account_ID.signin.aws.amazon.com/console/
```

Se crei un Account AWS alias per il tuo Account AWS ID, l'URL della pagina di accesso è simile all'esempio seguente.

```
https://Your_Account_Alias.signin.aws.amazon.com/console/
```

Considerazioni

- Account AWS Puoi avere un solo alias. Se crei un nuovo alias per l'account AWS , il nuovo alias sovrascrive l'alias precedente e l'URL contenente l'alias precedente smette di funzionare.
- L'alias dell' account deve includere solo cifre, lettere minuscole e trattini. Per ulteriori informazioni sulle limitazioni relative alle entità AWS dell'account, consultaIAM e AWS STS quote.
- L'alias dell'account deve essere univoco nei prodotti Amazon Web Services nella partizione di una determinata rete.

  Una partizione è un gruppo di AWS regioni. Ogni account AWS ha l'ambito di una partizione.

  Di seguito sono riportate le partizioni supportate:
  - `aws`- Regioni AWS
  - `aws-cn`: Regioni Cina
  - `aws-us-gov`- AWS GovCloud (US) Regioni

Creazione, eliminazione ed elenco di alias dell' Account AWS

Puoi utilizzare l' AWS Management Console API IAM o l'interfaccia a riga di comando per creare o eliminare il tuo Account AWS alias.

> **Note**
>
> Gli alias degli account non sono segreti e verranno visualizzati nell'URL della tua pagina di accesso pubblica. Non includere informazioni sensibili nell'alias del tuo account.
> L'URL originale contenente il tuo Account AWS ID rimane attivo e può essere utilizzato dopo aver creato l' Account AWS alias.

Creazione o modifica di un alias dell'account (console)

È possibile creare, modificare ed eliminare l'alias di un account dalla AWS Management Console.

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - `iam:ListAccountAliases`
> - `iam:CreateAccountAlias`

Per creare o modificare un alias dell'account (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Pannello di controllo.
3. Nella sezione Account AWS , accanto ad Alias dell'account, scegli Crea. Se l'alias esiste già, scegli Modifica.
4. Nella finestra di dialogo, inserisci il nome che desideri utilizzare per l'alias e quindi seleziona Salva modifiche.

> **Note**
>
> Puoi avere un solo alias associato al tuo Account AWS alla volta. Se crei un nuovo alias, l'alias precedente viene rimosso e l'URL di accesso associato all'alias precedente smette di funzionare.

Eliminazione di un alias dell'account (console)

Puoi eliminare l'alias dell'account dalla AWS Management Console.

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre almeno delle seguenti autorizzazioni:
>
> - `iam:ListAccountAliases`
> - `iam:CreateAccountAlias`

- `iam:DeleteAccountAlias`

Per eliminare un alias dell'account (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Pannello di controllo.
3. Nella sezione Account AWS , accanto ad Alias dell'account, scegli Elimina.

> **Note**
>
> L'unico URL di accesso per il tuo account si basa sull'ID account. Qualsiasi tentativo di connessione all'URL dell'alias non viene reindirizzato.

Creazione, eliminazione ed elenco di alias (AWS CLI)

> **Note**
>
> Per utilizzare i seguenti comandi, devi disporre almeno delle seguenti autorizzazioni IAM:
>
> - `iam:ListAccountAliases`
> - `iam:CreateAccountAlias`
> - `iam:DeleteAccountAlias`

Per creare un alias per l'URL della tua pagina di AWS Management Console accesso, esegui il seguente comando:

- `aws iam create-account-alias`

Per eliminare un alias Account AWS ID, esegui il comando seguente:

- `aws iam delete-account-alias`

Per visualizzare l'alias Account AWS ID, esegui il comando seguente:

- `aws iam list-account-aliases`

Example Comandi alias

Per visualizzare il tuo alias Account AWS ID, esegui il comando seguente.

```
$ aws iam list-account-aliases
{
    "AccountAliases": [
        "myaccountalias"
    ]
}
```

Per creare un alias per il tuo AWS Management Console accesso, esegui il comando seguente:

```
$ aws iam create-account-alias \
    --account-alias myaliasname
```

Se ha esito positivo, questo comando non produrrà alcun output.

Per eliminare un alias Account AWS ID, esegui il comando seguente.

```
$ aws iam delete-account-alias \
--account-alias myaliasname
```

Se ha esito positivo, questo comando non produrrà alcun output.

Creazione, eliminazione ed elenco di alias (API AWS)

> **Note**
>
> Per utilizzare le seguenti operazioni API, devi disporre almeno delle seguenti autorizzazioni IAM:
>
> - `iam:ListAccountAliases`
> - `iam:CreateAccountAlias`
> - `iam:DeleteAccountAlias`

Per creare un alias per l'URL della pagina di AWS Management Console accesso, richiama la seguente operazione:

- `CreateAccountAlias`

Per eliminare un alias Account AWS ID, richiama la seguente operazione:

- `DeleteAccountAlias`

Per visualizzare il tuo alias Account AWS ID, esegui la seguente operazione:

- `ListAccountAliases`

Nozioni di base su IAM

Usa questo tutorial per iniziare a usare AWS Identity and Access Management (IAM). Verrà descritto come creare ruoli, utenti e policy utilizzando la AWS Management Console.

AWS Identity and Access Management è una funzionalità che Account AWS offri senza costi aggiuntivi. Ti verrà addebitato solo l'utilizzo di altri AWS prodotti da parte dei tuoi utenti IAM. Per informazioni sui prezzi di altri AWS prodotti, consulta la pagina dei prezzi di Amazon Web Services.

> **Note**
>
> Questa serie di documenti è principalmente dedicata al servizio IAM. Per informazioni su come iniziare AWS e utilizzare più servizi per risolvere un problema, ad esempio la creazione e il lancio del primo progetto, consulta il Centro risorse per iniziare.

Indice

Prerequisiti

Prima di iniziare, devi accertarti di aver completato le fasi in Configurazione di IAM. Questo tutorial utilizza l'account amministratore che hai creato in quella procedura.

Creazione del primo utente IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Gli utenti possono essere organizzati in gruppi che condividono le stesse autorizzazioni.

> **Note**
>
> Come best practice di sicurezza, consigliamo di fornire l'accesso alle risorse tramite la federazione delle identità invece di creare utenti IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

Allo scopo di acquisire familiarità con il processo di creazione di un utente IAM, questo tutorial spiega come creare un utente IAM e un gruppo per l'accesso di emergenza.

Creazione del primo utente IAM

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console, seleziona il servizio IAM.
3. Nel pannello di navigazione seleziona Users (Utenti), quindi seleziona Add users (Aggiungi utenti).

   > **Note**
   >
   > Se hai abilitato IAM Identity Center, AWS Management Console viene visualizzato un promemoria che ti ricorda che è meglio gestire l'accesso degli utenti in IAM Identity Center. In questo tutorial, l'utente IAM che crei è destinato specificamente all'uso solo se non sono disponibili le credenziali dell'utente nel Centro identità IAM.

4. In Nome utente, inserisci **`EmergencyAccess`**. Il nome non può contenere spazi.
5. Seleziona la casella di controllo accanto a Fornisci l'accesso utente a AWS Management Console— opzionale, quindi scegli Voglio creare un utente IAM.
6. Per Console password (Password della console), seleziona Autogenerated password (Password generata automaticamente).
7. Seleziona la casella di controllo accanto a User must create a new password at next sign-in (recommended) (L'utente deve creare una nuova password all'accesso successivo [consigliato]). Poiché questo utente IAM è destinato all'accesso di emergenza, un amministratore attendibile ne conserverà la password e la fornirà solo quando necessario.

8. Nella pagina Set permissions (Imposta autorizzazioni), in Permissions options (Opzioni di autorizzazione), seleziona Add user to group (Aggiungi utente al gruppo). Quindi, in User groups (Gruppi di utenti), seleziona Create group (Crea gruppo).
9. Nella pagina Create user group (Crea gruppo di utenti), in User group name (Nome gruppo di utenti), inserisci **`EmergencyAccessGroup`**. Quindi, in Politiche di autorizzazione, seleziona AdministratorAccess.
10. Seleziona Create user group (Crea gruppo di utenti) per tornare alla pagina Set permissions (Imposta autorizzazioni).
11. In User groups (Gruppi di utenti), seleziona il nome del **`EmergencyAccessGroup`** creato in precedenza.
12. Seleziona Next (Successivo) per passare alla pagina Review and create (Rivedi e crea).
13. Nella pagina Review and create (Rivedi e crea), consulta l'elenco dei membri del gruppo di utenti da aggiungere al nuovo utente. Una volta pronto per continuare, seleziona Create user (Crea utente).
14. Nella pagina Retrieve password (Recupera password), seleziona Download .csv file (Scarica il file .csv) per salvare un file .csv con le informazioni sulle credenziali dell'utente (URL di connessione, nome utente e password).
15. Salva questo file per utilizzarlo se devi accedere a IAM e non hai accesso al tuo gestore di identità federato.

Il nuovo utente IAM viene visualizzato nell'elenco Users (Utenti). Seleziona il link User name (Nome utente) per visualizzare i dettagli dell'utente. In Summary (Riepilogo), copia l'ARN dell'utente negli appunti. Incolla l'ARN in un documento di testo in modo da poterlo utilizzare nella procedura successiva.

Creazione del primo ruolo

I ruoli IAM sono un modo sicuro per concedere autorizzazioni a entità di cui ti fidi. Un ruolo IAM presenta alcune analogie con un utente IAM. Ruoli e utenti sono entrambi principali con policy di autorizzazioni che determinano ciò che l'identità può o non può fare in AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo. L'utilizzo dei ruoli ti aiuta a seguire le best practice IAM. Puoi usare un ruolo per:

- Abilita le identità della forza lavoro e l'accesso all'utilizzo delle applicazioni abilitate a Identity Center. AWS Management Console AWS IAM Identity Center
- Delega l'autorizzazione a un AWS servizio per eseguire azioni per tuo conto.
- Abilita il codice applicativo in esecuzione su un'istanza Amazon EC2 per accedere o modificare le risorse AWS .
- Concedi l'accesso a un altro Account AWS.

> **Note**
>
> È possibile utilizzare AWS Identity and Access Management Roles Anywhere per consentire l'accesso alle identità delle macchine. L'utilizzo di IAM Roles Anywhere significa che non è necessario gestire credenziali a lungo termine per carichi di lavoro eseguiti all'esterno di. AWS Per ulteriori informazioni, consulta Cos'è AWS Identity and Access Management Roles Anywhere? nella Guida per l'utente di AWS Identity and Access Management Roles Anywhere.

IAM Identity Center e altri AWS servizi creano automaticamente ruoli per i propri servizi. Se utilizzi utenti IAM, ti consigliamo di creare ruoli che gli utenti possano assumere al momento dell'accesso. Ciò fornirà loro autorizzazioni provvisorie per la sessione anziché autorizzazioni a lungo termine.

La AWS Management Console procedura guidata che ti guida attraverso i passaggi per la creazione di un ruolo mostra passaggi leggermente diversi a seconda che tu stia creando un ruolo per un utente, un AWS servizio IAM o per un utente federato. L'accesso regolare all' Account AWS interno di un'organizzazione deve essere fornito utilizzando l'accesso federato. Se stai creando utenti IAM per scopi specifici, come l'accesso di emergenza o l'accesso programmatico, concedi a tali utenti IAM solo l'autorizzazione per assumere un ruolo e inserisci tali utenti IAM in gruppi specifici dei ruoli.

In questa procedura, crei un ruolo che fornisce SupportUser l'accesso all'utente EmergencyAccess IAM. Prima di iniziare la procedura, copia l'ARN dell'utente IAM negli appunti.

Creazione di un ruolo per un utente IAM

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console, seleziona il servizio IAM.
3. Nel pannello di navigazione della console IAM, scegli Ruoli e quindi Crea ruolo.

4. Scegli il tipo di ruolo Account AWS.
5. In Select trusted entity (Seleziona entità attendibile), in Trusted entity type (Tipo di entità attendibile), scegli Custom trust policy (Policy di attendibilità personalizzata).
6. Nella sezione Custom trust policy (Policy di attendibilità personalizzata), rivedi la policy di attendibilità di base. Questa è quella che useremo per questo ruolo. Utilizza l'editor Edit statement (Modifica istruzione) per aggiornare la policy di attendibilità:

   1. In Add actions for STS (Aggiungi operazioni per STS), seleziona Assume Role (Assumi ruolo).
   2. Accanto a Add a principal (Aggiungi un principale), seleziona Add (Aggiungi). Viene visualizzata la finestra Add principal (Aggiungi principale).

      In Principal type (Tipo di principale), seleziona IAM Users (Utenti IAM).

      In ARN, incolla l'ARN dell'utente IAM che hai copiato negli appunti.

      Seleziona Add principal (Aggiungi principale).
   3. Verifica che la riga `Principal` della policy di attendibilità contenga ora l'ARN specificato:

      ```
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/username" }
      ```
7. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).
8. In Add permissions (Aggiungi autorizzazioni), seleziona la casella di controllo accanto alla policy delle autorizzazioni da applicare. Per questo tutorial selezioneremo la politica di SupportUserfiducia. È quindi possibile utilizzare questo ruolo per risolvere e risolvere i problemi relativi a Account AWS e aprire casi di supporto con. AWS Al momento non fisseremo un limite delle autorizzazioni.
9. Seleziona Successivo.
10. In Name, review, and create (Nomina, verifica e crea) inserisci le seguenti informazioni:

    - Per Nome ruolo, inserisci un nome che identifichi questo ruolo, ad esempio. SupportUserRole
    - In Description (Descrizione), spiega l'uso previsto del ruolo.

    Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.
11. Seleziona Create role (Crea ruolo).

Una volta creato il ruolo, condividi le informazioni del ruolo con le persone che lo richiedono. Puoi condividere le informazioni sul ruolo nei seguenti modi:

- Role link (Link del ruolo): invia agli utenti un collegamento che indirizza alla pagina Switch Role (Cambia ruolo) con tutti i dettagli già compilati.
- ID account o alias: fornisci a ciascun utente il nome del ruolo insieme al numero dell'ID account o all'alias dell'account. L'utente accede quindi alla pagina Switch Role (Cambia ruolo) e aggiunge i dettagli manualmente.
- Salvataggio delle informazioni sul collegamento al ruolo insieme alle credenziali dell' EmergencyAccess utente.

Per informazioni dettagliate, vedi Fornire informazioni all'utente.

Creazione della prima policy IAM

Le policy IAM sono collegate alle identità (utenti, gruppi di utenti o ruoli) o alle risorse AWS . Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni.

Creazione della prima policy IAM

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console, seleziona il servizio IAM.
3. Nel pannello di navigazione, selezionare Policy.

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
4. Scegli Crea policy.
5. Nella pagina Crea policy, scegli Operazioni, quindi scegli Importa policy.
6. Nella finestra Importa policy, nella casella Trova policy, digita **`power`** per ridurre l'elenco di policy. Seleziona la PowerUserAccesspolitica.
7. Scegli Importa policy. La policy viene visualizzata nella scheda JSON.
8. Seleziona Successivo.

9. Nella pagina Rivedi e crea, digita **`PowerUserExamplePolicy`** in Nome della policy. In Description (Descrizione), digitare **`Allows full access to all services except those for user management`**. Quindi selezionare Create policy (Crea policy) per salvare la policy.

È possibile collegare questa policy a un ruolo per fornire agli utenti che assumono quel ruolo le autorizzazioni ad essa associate. La PowerUserAccesspolitica viene comunemente utilizzata per fornire l'accesso agli sviluppatori.

Accesso programmatico

Gli utenti necessitano di un accesso programmatico se desiderano interagire con l' AWS AWS Management Console esterno di. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede a: AWS

- Se gestisci le identità in IAM Identity Center, le AWS API richiedono un profilo e AWS Command Line Interface richiedono un profilo o una variabile di ambiente.
- Se hai utenti IAM, le AWS API e le AWS Command Line Interface richiedono chiavi di accesso. Quando possibile, creare credenziali temporanee formate da un ID della chiave di accesso, una chiave di accesso segreta e un token di sicurezza che ne indica la scadenza.

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali a breve termine per firmare le richieste programmatiche alle AWS CLI o AWS API (direttamente o utilizzando gli SDK). AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare: • Per farlo AWS CLI, segui le istruzioni in Ottenere le credenziali del ruolo IAM per l'accesso alla CLI nella Guida per AWS IAM Identity Center l'utente.

Quale utente necessita dell'accesso programmatico?	Per	Come
		• Per le AWS API, segui le istruzioni contenute nelle credenziali SSO nella Guida di riferimento agli AWS SDK e agli strumenti.
IAM	Utilizza credenziali a breve termine per firmare le richieste programmatiche alle AWS API AWS CLI o (direttamente o utilizzando gli SDK). AWS	Segui le istruzioni riportate in Utilizzo delle credenziali temporanee con le risorse. AWS
IAM	Utilizza credenziali a lungo termine per firmare le richieste programmatiche alle AWS CLI o AWS API (direttamente o utilizzando gli SDK). AWS (Non consigliato)	Segui le istruzioni in Gestione delle chiavi di accesso per gli utenti IAM.

Best practice per la sicurezza e casi d'uso in AWS Identity and Access Management

AWS Identity and Access Management (IAM) offre una serie di funzionalità di sicurezza da prendere in considerazione durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

Per ottenere i vantaggi maggiori da IAM, leggi con attenzione le best practice consigliate. Un modo per fare ciò è verificare in che modo IAM viene utilizzato negli scenari reali per funzionare con altri servizi AWS .

Argomenti

Best practice per la sicurezza in IAM

Follow us on Twitter

Le AWS Identity and Access Management migliori pratiche sono state aggiornate il 14 luglio 2022.

Per proteggere AWS le tue risorse, segui queste best practice per AWS Identity and Access Management (IAM).

Argomenti

- Segui le best practice per proteggere le credenziali di utente root
- Assegna le autorizzazioni con privilegi minimi
- Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi
- Utilizzare IAM Access Analyzer per generare policy con privilegi minimi in base all'attività di accesso
- Esaminare e rimuovere regolarmente utenti, ruoli, autorizzazioni, criteri e credenziali inutilizzati
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso
- Verifica dell'accesso multi-account e pubblico alle risorse con IAM Access Analyzer
- Usa IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali
- Stabilisci guardrail delle autorizzazioni su più account
- Utilizzare i limiti delle autorizzazioni per delegare la gestione delle autorizzazioni all'interno di un account

Richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee

Utenti umani, noti anche come identità umane, sono le persone, gli amministratori, gli sviluppatori, gli operatori e i consumatori delle tue applicazioni. Devono avere un'identità per accedere agli AWS ambienti e alle applicazioni dell'utente. Gli utenti umani membri dell'organizzazione sono noti anche come identità della forza lavoro. Gli utenti umani possono anche essere utenti esterni con cui collabori e che interagiscono con le tue risorse AWS . Possono farlo tramite un browser Web, un'applicazione client, un'app mobile o strumenti interattivi della riga di comando.

Richiedi ai tuoi utenti umani di utilizzare credenziali temporanee per l'accesso AWS. Puoi utilizzare un provider di identità per consentire agli utenti umani di fornire un accesso federato Account AWS assumendo ruoli che forniscono credenziali temporanee. Per una gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center (IAM Identity Center) per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Puoi gestire le tue identità utente con IAM Identity Center o gestire le autorizzazioni di accesso per le identità degli utenti in IAM Identity Center da un provider di identità esterno. Per ulteriori informazioni, consulta Che cos'è AWS IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Per ulteriori informazioni sui ruoli, consulta Termini e concetti dei ruoli.

Richiedi ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per accedere AWS

Un carico di lavoro è una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione o un processo back-end. Il carico di lavoro può avere applicazioni, strumenti operativi e componenti che richiedono un'identità per effettuare richieste a Servizi AWS, ad esempio richieste di lettura dei dati. Queste identità includono macchine in esecuzione nei tuoi AWS ambienti, come AWS Lambda istanze o funzioni di Amazon EC2.

Puoi anche gestire identità computer per soggetti esterni che necessitano di accesso. Per concedere l'accesso alle identità computer, puoi utilizzare i ruoli IAM. I ruoli IAM dispongono di autorizzazioni specifiche e forniscono un modo per accedere AWS affidandosi a credenziali di sicurezza temporanee con una sessione di ruolo. Inoltre, potresti avere macchine esterne AWS che richiedono l'accesso ai tuoi ambienti. AWS Per i computer che funzionano all'esterno dell' AWS utente, è possibile utilizzare AWS Identity and Access Management Roles Anywhere. Per ulteriori informazioni sui ruoli, consulta Ruoli IAM. Per informazioni dettagliate su come utilizzare i ruoli per delegare l'accesso da una parte all'altra Account AWS, consultaTutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.

Richiedere l'autenticazione a più fattori (MFA)

Ti consigliamo di utilizzare i ruoli IAM per utenti umani e carichi di lavoro che accedono alle tue risorse AWS in modo che utilizzino credenziali temporanee. Tuttavia, per gli scenari in cui hai bisogno di utenti IAM o root nel tuo account, richiedi MFA per una maggiore sicurezza. Con MFA, gli utenti dispongono di un dispositivo che genera una risposta a una richiesta di autenticazione. Per completare la procedura di accesso, sono necessarie le credenziali dell'utente e la risposta generata dal dispositivo. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

Se utilizzi IAM Identity Center per la gestione centralizzata degli accessi per utenti umani, puoi utilizzare le funzionalità MFA di IAM Identity Center quando la tua origine di identità è configurata con IAM Identity Center Identity Store, AWS Managed Microsoft AD o AD Connector. Per ulteriori informazioni sull'MFA, in IAM Identity Center consulta Autenticazione a più fattori (MFA) nella Guida per l'utente di AWS IAM Identity Center .

Aggiornamento delle chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine

Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare credenziali a lungo termine come le chiavi di accesso. Tuttavia, per gli scenari in cui sono necessari utenti IAM con accesso a livello di programmazione e credenziali a lungo termine, si consiglia di aggiornare le chiavi di accesso all'occorrenza, ad esempio quando un dipendente lascia l'azienda. Ti consigliamo di utilizzare informazioni utilizzate per l'ultimo accesso IAM per aggiornare e rimuovere le chiavi di accesso in modo sicuro. Per ulteriori informazioni, consulta Aggiornamento delle chiavi di accesso.

Esistono casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM in AWS. Alcuni dei casi d'uso sono i seguenti:

- Carichi di lavoro che non possono utilizzare ruoli IAM: è possibile eseguire un carico di lavoro da una posizione che deve accedere a AWS. In alcune situazioni, non è possibile utilizzare i ruoli IAM per fornire credenziali temporanee, ad esempio per i plug-in. WordPress In queste situazioni, per autenticarti a AWS usa le chiavi di accesso a lungo termine dell'utente IAM per quel carico di lavoro.
- AWS Client di terze parti: se utilizzi strumenti che non supportano l'accesso con IAM Identity Center, come AWS client o fornitori di terze parti che non sono ospitati su AWS, utilizza le chiavi di accesso a lungo termine degli utenti IAM.
- AWS CodeCommit accesso: se utilizzi CodeCommit per archiviare il codice, puoi utilizzare un utente IAM con chiavi SSH o credenziali specifiche del servizio CodeCommit per l'autenticazione nei tuoi repository. Si consiglia di eseguire questa operazione oltre a utilizzare un utente di IAM Identity Center per l'autenticazione normale. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue o alle tue applicazioni cloud. Account AWS Per consentire agli utenti di accedere ai tuoi CodeCommit repository senza configurare gli utenti IAM, puoi configurare l'utilità. git-remote-codecommit Per ulteriori informazioni su IAM e CodeCommit, consulta. Usare IAM con CodeCommit: credenziali Git, chiavi SSH e AWS chiavi di accesso Per ulteriori informazioni sulla configurazione dell'git-remote-codecommitutilità, consulta Connessione ai AWS CodeCommit repository con credenziali rotanti nella Guida per l'utente.AWS CodeCommit
- Accesso ad Amazon Keyspaces (per Apache Cassandra): in una situazione in cui non è possibile utilizzare gli utenti in IAM Identity Center, ad esempio per scopi di test per la compatibilità con Cassandra, puoi utilizzare un utente IAM con credenziali specifiche del servizio per l'autenticazione con Amazon Keyspaces. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro

che hanno bisogno di accedere alle tue applicazioni Account AWS o alle tue applicazioni cloud. Puoi anche connetterti ad Amazon Keyspaces utilizzando credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di credenziali temporanee per connettersi ad Amazon Keyspaces utilizzando un ruolo IAM e il plugin SIGv4 nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Segui le best practice per proteggere le credenziali di utente root

Quando crei un file Account AWS, stabilisci le credenziali dell'utente root per accedere a. AWS Management Console Proteggi le tue credenziali utente root nello stesso modo in cui proteggeresti altre informazioni personali sensibili. Per comprendere meglio come proteggere e dimensionare i processi degli utenti root, consulta Best practice per gli utenti root per Account AWS.

Assegna le autorizzazioni con privilegi minimi

Quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Potresti iniziare con autorizzazioni generiche mentre esplori le autorizzazioni necessarie per il tuo carico di lavoro o il caso d'uso. Man mano che il tuo caso d'uso matura, puoi lavorare per ridurre le autorizzazioni concesse per lavorare con il privilegio minimo. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM.

Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi

Per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite di AWS che concedono autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per l'uso da parte di tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo Policy gestite dal cliente specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta AWS politiche gestite. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, consulta. AWS politiche gestite per le funzioni lavorative

Utilizzare IAM Access Analyzer per generare policy con privilegi minimi in base all'attività di accesso

Per concedere solo le autorizzazioni richieste per eseguire un'attività, puoi generare policy in funzione dell'attività di accesso che hai effettuato l'accesso in AWS CloudTrail. IAM Access Analyzer analizza i servizi e le azioni utilizzati dai tuoi ruoli IAM e quindi genera una policy dettagliata che puoi utilizzare. Dopo aver testato ogni policy generata, puoi distribuirla nell'ambiente di produzione. In questo modo si garantisce di concedere solo le autorizzazioni necessarie ai carichi di lavoro. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Esaminare e rimuovere regolarmente utenti, ruoli, autorizzazioni, criteri e credenziali inutilizzati

Potresti avere utenti, ruoli, autorizzazioni, policy o credenziali IAM che non servono più nel tuo Account AWS. IAM fornisce le ultime informazioni di accesso per aiutarti a identificare gli utenti, i ruoli, le autorizzazioni, le policy e le credenziali che non ti servono più in modo da poterli rimuovere. In questo modo puoi ridurre il numero di utenti, ruoli, autorizzazioni, criteri e credenziali da monitorare. È possibile utilizzare queste informazioni per perfezionare le policy IAM e aderire meglio al principio del privilegio minimo. Per ulteriori informazioni, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso

È possibile specificare le condizioni che stabiliscono che una dichiarazione di policy è attiva. In questo modo, è possibile concedere l'accesso ad azioni e risorse, ma solo se la richiesta di accesso soddisfa condizioni specifiche. Ad esempio, è possibile scrivere una condizione politica per specificare che tutte le richieste devono essere inviate utilizzando SSL. È inoltre possibile utilizzare le condizioni per concedere l'accesso alle azioni di servizio, ma solo se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Condition.

Verifica dell'accesso multi-account e pubblico alle risorse con IAM Access Analyzer

Prima di concedere le autorizzazioni per l'accesso pubblico o su più account AWS, ti consigliamo di verificare se tale accesso è richiesto. Puoi utilizzare IAM Access Analyzer per visualizzare in anteprima e analizzare l'accesso multi-account e pubblico per i tipi di risorse supportati. Puoi farlo

esaminando i risultati generati da IAM Access Analyzer. Questi risultati consentono di verificare che i controlli di accesso alle risorse garantiscano l'accesso previsto. Inoltre, quando aggiorni le autorizzazioni pubbliche e multi-account, puoi verificare l'effetto delle modifiche prima di distribuire nuovi controlli di accesso alle tue risorse. Inoltre, IAM Access Analyzer monitora continuamente i tipi di risorse supportati e genera un risultato per le risorse che consentono l'accesso multi-account o pubblico. Per ulteriori informazioni, consulta Anteprima dell'accesso con API IAM Access Analyzer.

Usa IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali

Convalida le policy che crei per assicurarti che aderiscano al Linguaggio policy IAM (JSON) e best practice di IAM. È possibile convalidare le policy utilizzando la validazione delle policy di IAM Access Analyzer. IAM Access Analyzer fornisce oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Durante la creazione di nuove policy o la modifica di policy esistenti nella console, IAM Access Analyzer fornisce suggerimenti per aiutarti a perfezionare e convalidare le policy prima di salvarle. Inoltre, consigliamo di rivedere e convalidare tutte le policy esistenti. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer. Per ulteriori informazioni sui controlli delle policy di IAM Access Analyzer, consulta Documentazione di riferimento sui controlli delle policy di IAM Access Analyzer.

Stabilisci guardrail delle autorizzazioni su più account

Man mano che ridimensioni i carichi di lavoro, separali utilizzando più account gestiti con. AWS Organizations Ti consigliamo di utilizzare le Policy di controllo dei servizi(SCP) di Organizations per stabilire i guardrail delle autorizzazioni per controllare l'accesso per tutti gli utenti e i ruoli IAM nei tuoi account. Gli SCP sono un tipo di politica organizzativa che è possibile utilizzare per gestire le autorizzazioni all'interno dell'organizzazione a livello di AWS organizzazione, unità organizzativa o account. I guardrail delle autorizzazioni stabilite dall'utente si applicano a tutti gli utenti e ai ruoli degli account coperti. Tuttavia, le SCP da sole non sono sufficienti a concedere le autorizzazioni agli account nella tua organizzazione. Per fare questo, l'amministratore deve comunque collegare policy basate su identità o policy basate su risorse agli utenti o ai ruoli IAM o alle risorse degli account. Per ulteriori informazioni, consulta AWS Organizations, account e guardrail IAM.

Utilizzare i limiti delle autorizzazioni per delegare la gestione delle autorizzazioni all'interno di un account

In alcuni scenari, è possibile che tu intenda delegare la gestione delle autorizzazioni all'interno di un account ad altri. Ad esempio, potresti consentire agli sviluppatori di creare e gestire ruoli per

i loro carichi di lavoro. Quando deleghi le autorizzazioni ad altri, usa Limiti delle autorizzazioni per impostare le autorizzazioni massime delegate. Un limite delle autorizzazioni è una funzione avanzata per l'utilizzo di una policy gestita per impostare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un ruolo IAM. Il limite delle autorizzazioni non concedere l'accesso di per sé. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM.

Best practice per gli utenti root per Account AWS

La prima volta che ne crei un Account AWS, inizi con un set predefinito di credenziali con accesso completo a tutte le AWS risorse del tuo account. Questa identità è chiamata utente root di Account AWS. Ti consigliamo vivamente di non accedere all'utente Account AWS root a meno che tu non abbia un'attività che richiede le credenziali dell'utente root. È necessario proteggere le credenziali dell'utente root e i meccanismi di ripristino dell'account per evitare di esporre le proprie credenziali altamente privilegiate per usi non autorizzati.

Invece di accedere all'utente root, crea un utente amministrativo per le attività quotidiane.

- Per una singola versione autonoma Account AWS, vedi. Crea un utente con accesso amministrativo
- Per più utenti Account AWS gestiti AWS Organizations, consulta Configurare Account AWS l'accesso per un utente amministrativo di IAM Identity Center.

Con il tuo utente amministrativo, puoi quindi creare identità aggiuntive per gli utenti che necessitano di accedere alle risorse del tuo Account AWS. Ti consigliamo vivamente di richiedere agli utenti di autenticarsi con credenziali temporanee al momento dell'accesso. AWS

- Utilizzala singolarmente, autonoma Account AWS, Ruoli IAM per creare identità nel tuo account con autorizzazioni specifiche. I ruoli sono destinai a essere assunti da chiunque ne abbia bisogno. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo. A differenza dei ruoli IAM, Utenti IAM dispongono di credenziali a lungo termine come password e chiavi di accesso. Ove possibile, le best practice raccomandano di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso.
- Per più utenti Account AWS gestiti tramite Organizations, utilizza gli utenti della forza lavoro di IAM Identity Center. Con IAM Identity Center, puoi gestire centralmente gli utenti Account AWS e le autorizzazioni relative a tali account. Gestisci le identità degli utenti con IAM Identity Center o

con un provider di identità esterno. Per ulteriori informazioni, consulta Che cos'è AWS IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Argomenti

Proteggi le credenziali di utente root per impedirne l'uso non autorizzato

Proteggi le credenziali dell'utente root e usale solo per le attività che le richiedono. Per prevenire l'uso non autorizzato, non condividere la password dell'utente root, l'MFA, le chiavi di accesso, le coppie di chiavi CloudFront o i certificati di firma con nessuno, ad eccezione di coloro che hanno esigenze aziendali rigorose per accedere all'utente root.

Non memorizzate la password dell'utente root con strumenti che dipendono da un account Servizi AWS a cui si accede utilizzando la stessa password. Se perdi o dimentichi la password dell'utente root, non potrai accedere a questi strumenti. Si consiglia di dare priorità alla resilienza e di richiedere a due o più persone di autorizzare l'accesso alla posizione di archiviazione. L'accesso alla password o alla sua posizione di archiviazione deve essere registrato e monitorato.

Utilizza una password dell'utente root sicura per proteggere l'accesso

Consigliamo di utilizzare una password complessa e univoca. Strumenti come i gestori di password con potenti algoritmi di generazione di password possono aiutarti a raggiungere questi obiettivi. AWS richiede che la password soddisfi le seguenti condizioni:

- Deve avere un minimo di 8 caratteri e un massimo di 128 caratteri.

- Deve includere almeno tre dei seguenti tipi di caratteri: maiuscole, minuscole, numeri e i simboli ! @ # $ % ^ & * () <> [] {} | _+-=.
- Non deve essere identica al tuo Account AWS nome o indirizzo email.

Per ulteriori informazioni, consulta Cambiare la password per Utente root dell'account AWS.

Abilita l'autenticazione a più fattori (MFA) per la sicurezza dell'utente root

Poiché un utente root può eseguire azioni privilegiate, è fondamentale aggiungere MFA per l'utente root come secondo fattore di autenticazione oltre all'indirizzo e-mail e alla password come credenziali di accesso. Ti consigliamo vivamente di abilitare più MFA per le credenziali dell'utente root per fornire maggiore flessibilità e resilienza nella tua strategia di sicurezza. È possibile registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con l'utente root Account AWS .

- Le chiavi di sicurezza hardware certificate FIDO sono fornite da fornitori terzi. Per ulteriori informazioni, vedere Abilitare una chiave di sicurezza FIDO per l'utente Account AWS root.
- Token TOTP hardware: un dispositivo hardware che genera un codice numerico a sei cifre basato sull'algoritmo TOTP (password monouso). Per ulteriori informazioni, vedere Abilitare un token TOTP hardware per l'utente Account AWS root.
- Un'applicazione di autenticazione virtuale che viene eseguita su un telefono o altro dispositivo e simula un dispositivo fisico. Per ulteriori informazioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root.

Non creare chiavi di accesso per l'utente root

Le chiavi di accesso consentono di eseguire comandi nell'interfaccia a riga di AWS comando (AWS CLI) o utilizzare le operazioni API da uno degli AWS SDK. Ti consigliamo vivamente di non creare coppie di chiavi di accesso per l'utente root, poiché l'utente root ha pieno accesso a tutte Servizi AWS le risorse dell'account, incluse le informazioni di fatturazione.

Poiché solo alcune attività richiedono l'utilizzo dell'utente root e in genere le esegui di rado, consigliamo di accedere a per eseguire le AWS Management Console attività dell'utente root. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Utilizza l'approvazione di più persone per l'accesso come utente root laddove possibile

Valuta la possibilità di utilizzare l'approvazione di più persone per garantire che nessuna persona possa accedere sia all'MFA che alla password per l'utente root. Alcune aziende aggiungono un ulteriore livello di sicurezza configurando un gruppo di amministratori con accesso alla password e un altro gruppo di amministratori con accesso alla MFA. Per eseguire l'accesso utilizzando le credenziali dell'utente root è necessario che si riuniscano due membri, uno di ciascun gruppo.

Usa un indirizzo email di gruppo per le credenziali dell'utente root

Utilizza un indirizzo e-mail gestito dalla tua azienda e inoltra i messaggi ricevuti direttamente a un gruppo di utenti. Se è AWS necessario contattare il proprietario dell'account, questo approccio riduce il rischio di ritardi nella risposta, anche se le persone sono in vacanza, sono in malattia o hanno lasciato l'attività. L'indirizzo e-mail utilizzato per l'utente root non deve essere utilizzato per altri scopi.

Limita l'accesso ai meccanismi di recupero dell'account

Assicurati di sviluppare un processo per gestire i meccanismi di recupero delle credenziali degli utenti root nel caso in cui sia necessario accedervi in caso di emergenza, come l'acquisizione del tuo account amministrativo.

- Assicurati di avere accesso alla casella di posta elettronica dell'utente root in modo da poter reimpostare una password utente root persa o dimenticata.
- Se la MFA per l'utente Account AWS root viene persa, danneggiata o non funziona, è possibile accedere utilizzando un'altra MFA registrata con le stesse credenziali dell'utente root. Se hai perso l'accesso a tutte le tue MFA, hai bisogno sia del numero di telefono che dell'indirizzo e-mail utilizzati per registrare il tuo account, per essere aggiornati e accessibili per recuperare la tua MFA. Per i dettagli, consulta Recupero di un dispositivo MFA per utenti root.
- Se scegli di non memorizzare la password dell'utente root e l'MFA, il numero di telefono registrato nell'account può essere utilizzato come metodo alternativo per recuperare le credenziali dell'utente root. Assicurati di avere accesso al numero di telefono di contatto, mantieni aggiornato il numero di telefono e limita l'accesso alla gestione del numero di telefono.

Nessuno dovrebbe avere accesso sia alla casella di posta elettronica che al numero di telefono, poiché entrambi sono canali di verifica per recuperare la password dell'utente root. È importante che due gruppi di persone gestiscano questi canali. Un gruppo ha accesso al tuo indirizzo email

principale e un altro gruppo che ha accesso al numero di telefono principale per recuperare l'accesso al tuo account come utente root.

Proteggi le credenziali utente root dell'account Organizations

Quando si passa a una strategia multi-account con Organizations, ognuno di voi Account AWS dispone delle proprie credenziali utente root che dovete proteggere. L'account che usi per creare la tua organizzazione è l'account di gestione e gli altri account dell'organizzazione sono account membro.

Proteggi le credenziali utente root per gli account dei membri

Se utilizzi Organizations per gestire più account, puoi adottare due strategie per proteggere l'accesso degli utenti root nelle tue Organizzazioni.

- Proteggi le credenziali utente root dell'account Organizations con MFA.
- Non reimpostate la password dell'utente root per i vostri account e recuperate l'accesso ad essa solo quando necessario utilizzando la procedura di reimpostazione della password. Quando crei un account membro nella tua organizzazione, Organizations crea automaticamente un ruolo IAM nell'account membro che consente all'account di gestione l'accesso temporaneo all'account membro.

Per i dettagli, consulta Accedere agli account dei membri della tua organizzazione nella Guida per l'utente di Organizations.

Imposta controlli di sicurezza preventivi nelle Organizzazioni utilizzando una policy di controllo dei servizi (SCP)

Se utilizzi Organizations per gestire più account, puoi applicare un SCP per limitare l'accesso all'utente root dell'account membro. Negare tutte le azioni degli utenti root negli account dei membri, ad eccezione di alcune azioni di tipo root, aiuta a prevenire l'accesso non autorizzato. Per i dettagli, consulta utilizza una SCP per limitare ciò che le operazioni che un utente root nei tuoi account membri può eseguire.

Monitora l'accesso e l'utilizzo

Ti consigliamo di utilizzare gli attuali meccanismi di tracciamento per monitorare, avvisare e segnalare l'accesso e l'uso delle credenziali dell'utente root, compresi gli avvisi che annunciano

l'accesso e l'utilizzo dell'utente root. I seguenti servizi possono contribuire a garantire che l'utilizzo delle credenziali dell'utente root sia monitorato ed eseguire controlli di sicurezza che possono aiutare a prevenire l'uso non autorizzato.

- Se desideri ricevere notifiche sull'attività di accesso dell'utente root nel tuo account, puoi sfruttare Amazon CloudWatch per creare una regola Events che rileva quando vengono utilizzate le credenziali dell'utente root e attiva una notifica al tuo amministratore della sicurezza. Per i dettagli, consulta Monitora e invia notifiche sull'attività degli utenti root. Account AWS
- Se desideri configurare notifiche per avvisarti delle azioni approvate degli utenti root, puoi sfruttare Amazon EventBridge insieme ad Amazon SNS per scrivere EventBridge una regola per tenere traccia dell'utilizzo degli utenti root per l'azione specifica e inviarti notifiche utilizzando un argomento di Amazon SNS. Per un esempio, consulta Inviare una notifica quando viene creato un oggetto Amazon S3.
- Se lo utilizzi già GuardDuty come servizio di rilevamento delle minacce, puoi estenderne la capacità di avvisarti quando le credenziali degli utenti root vengono utilizzate nel tuo account.

Gli avvisi dovrebbero includere, ma non esclusivamente, l'indirizzo e-mail utilizzato per l'utente root stesso. Controlla che ci siano delle prassi attive perché il personale che riceve un avviso di questo tipo comprenda come convalidare che è previsto l'accesso utente root e come sottoporre la questione ai livelli gerarchici superiori se ritiene che sia in corso un incidente di sicurezza. Per un esempio su come configurare gli avvisi, consulta Monitoraggio e notifiche sull'attività dell'utente root Account AWS.

Valuta la conformità con l'MFA per l'utente root

- AWS Config utilizza regole per aiutare a far rispettare le migliori pratiche per gli utenti root. È possibile utilizzare regole AWS gestite per richiedere agli utenti root di abilitare l'autenticazione a più fattori (MFA). AWS Config può anche identificare le chiavi di accesso per l'utente root.
- Security Hub offre una visione completa dello stato di sicurezza in uso AWS e aiuta a valutare l' AWS ambiente rispetto agli standard e alle best practice del settore della sicurezza, ad esempio avere l'MFA sull'utente root e non avere chiavi di accesso per l'utente root. Per i dettagli sulle regole disponibili, consulta AWS Identity and Access Management i controlli nella Guida per l'utente di Security Hub.
- Trusted Advisor fornisce un controllo di sicurezza per sapere se l'MFA non è abilitata sull'account utente root. Per ulteriori informazioni, consulta MFA sull'account root nella Guida per l'utente di AWS .

Se devi segnalare un problema di sicurezza sul tuo account, consulta Segnalazione di e-mail sospette o Segnalazione di vulnerabilità. In alternativa, puoi contattare AWS per ricevere assistenza e indicazioni aggiuntive.

Casi d'uso di business per IAM

Un semplice caso d'uso aziendale per IAM può aiutarti a comprendere i modi di base per implementare il servizio per controllare l' AWS accesso degli utenti. Il caso d'uso viene descritto in termini generali, senza i meccanismi del modo in cui si desidera utilizzare l'API IAM per ottenere i risultati desiderati.

Questo caso d'uso esamina due modi tipici in cui un'azienda fittizia chiamata Example Corp può utilizzare IAM. Il primo scenario considera Amazon Elastic Compute Cloud (Amazon EC2). Il secondo considera Amazon Simple Storage Service (Amazon S3).

Per ulteriori informazioni sull'utilizzo di IAM con altri servizi di AWS, consultaAWS servizi che funzionano con IAM.

Argomenti

Configurazione iniziale di Example Corp

Nikki Wolf e Mateo Jackson sono i fondatori di Example Corp. Dopo aver avviato l'azienda, creano un Account AWS e configurato AWS IAM Identity Center(IAM Identity Center) per creare account amministrativi da utilizzare con le proprie risorse. AWS Quando si configura l'accesso all'account per l'utente amministrativo, il Centro identità IAM crea un ruolo IAM corrispondente. Questo ruolo, controllato da IAM Identity Center, viene creato nel pertinente Account AWS e le politiche specificate nel set di AdministratorAccessautorizzazioni sono allegate al ruolo.

Poiché ora dispongono di account di amministratore, Nikki e Mateo non devono più utilizzare il proprio utente root per accedere all' Account AWS. Pianificano l'uso dell'utente root solo per completare le attività che possono essere eseguite soltanto dall'utente root. Dopo aver esaminato le best practice di sicurezza, configurano l'autenticazione a più fattori MFA per le credenziali dell'utente root e decidono come proteggere tali credenziali.

Man mano che l'azienda cresce, assume dipendenti che lavorano come sviluppatori, amministratori, tester, manager e amministratori di sistema. Nikki è responsabile delle operazioni, mentre Mateo gestisce i team di ingegneria. Hanno creato un server di dominio Active Directory per gestire gli account dei dipendenti e gestire l'accesso alle risorse interne dell'azienda.

Per consentire ai dipendenti di accedere alle AWS risorse, utilizzano IAM Identity Center per connettere Active Directory della propria azienda ai propri Account AWS.

Poiché hanno collegato Active Directory al Centro identità IAM, gli utenti, il gruppo e l'appartenenza al gruppo vengono sincronizzati e definiti. Devono assegnare set di autorizzazioni e ruoli ai diversi gruppi per offrire agli utenti il livello corretto di accesso alle AWS risorse. AWS politiche gestite per le funzioni lavorativeUtilizzano AWS Management Console per creare questi set di autorizzazioni:

- Amministratore
- Fatturazione
- Sviluppatori
- Amministratori di rete
- Amministratori di database
- Amministratori di sistema
- Utenti del gruppo Supporto

Quindi assegnano i set di autorizzazioni ai ruoli assegnati ai rispettivi gruppi di Active Directory.

Per una step-by-step guida che descrive la configurazione iniziale di IAM Identity Center, consulta Guida introduttiva nella Guida per l'AWS IAM Identity Center utente. Per ulteriori informazioni sul provisioning dell'accesso utente del Centro identità IAM, consulta Accesso Single Sign-on agli account AWS nella Guida per l'utente di AWS IAM Identity Center .

Caso d'uso per IAM con Amazon EC2

Un'azienda come Example Corp normalmente utilizza IAM per interagire con servizi come Amazon EC2. Per capire questa parte del caso d'uso, è necessaria una conoscenza di base di Amazon EC2. Per ulteriori informazioni su Amazon EC2, consulta la Amazon EC2 User Guide.

Autorizzazioni Amazon EC2 per i gruppi di utenti

Per fornire un controllo «perimetrale», Nikki attribuisce una policy al gruppo di utenti. AllUsers Questa politica nega qualsiasi AWS richiesta da parte di un utente se l'indirizzo IP di origine è esterno alla rete aziendale di Example Corp.

Presso Example Corp, gruppi diversi richiedono autorizzazioni diverse:

- Amministratori di sistema: è necessaria l'autorizzazione per creare e gestire le AMI, le istanze, gli snapshot, i volumi, i gruppi di sicurezza e così via. Nikki allega la policy `AmazonEC2FullAccess` AWS gestita al gruppo di SysAdmins utenti che concede ai membri del gruppo l'autorizzazione a utilizzare tutte le azioni di Amazon EC2.
- Sviluppatori: è necessaria la possibilità di collaborare solo con le istanze. Mateo quindi crea e collega una policy al gruppo di utenti Sviluppatori che consente agli sviluppatori di chiamare `DescribeInstances`, `RunInstances`, `StopInstances`, `StartInstances` e `TerminateInstances`.

  > **Note**
  >
  > Amazon EC2 utilizza chiavi SSH, password Windows e gruppi di sicurezza per controllare chi ha accesso al sistema operativo di istanze Amazon EC2 specifiche. Non esiste un metodo nel sistema IAM per consentire o rifiutare l'accesso al sistema operativo di una determinata istanza.

- Utenti del gruppo Supporto: non devono essere in grado di eseguire operazioni Amazon EC2 eccetto elencare risorse Amazon EC2 correntemente disponibili. Pertanto, Nikki crea e collega una policy al gruppo di utenti Supporto che consente di chiamare solo le operazioni API "Describe" di Amazon EC2.

Per esempi di come potrebbero essere queste policy, consulta Esempi di policy basate su identità IAM e Using AWS Identity and Access Management nella Amazon EC2 User Guide.

Modifica della funzione lavorativa dell'utente

A un certo punto, uno degli sviluppatori, Paulo Santos, cambia le funzioni lavorative e diventa un responsabile. In qualità di responsabile, Paulo entra a far parte del gruppo di utenti Supporto in modo da poter aprire casi di supporto per i suoi sviluppatori. Mateo sposta Paulo dal gruppo di utenti Sviluppatori al gruppo di utenti Supporto. Come risultato di questa mossa, la sua possibilità di

interagire con le istanze Amazon EC2 è limitata. Non può avviare istanze. Inoltre, non può arrestare o terminare le istanze esistenti, anche se era l'utente che ha avviato l'istanza. Può elencare solo le istanze che gli utenti di Example Corp hanno lanciato.

Caso d'uso per IAM con Amazon S3

Le aziende come Example Corp in genere utilizzano IAM anche con Amazon S3. John ha creato un bucket Amazon S3 per l'azienda chiamato aws-s3-bucket.

Creazione di altri utenti e gruppi di utenti

Come dipendenti, Zhang Wei e Mary Major devono essere in grado di creare i propri dati nel bucket dell'azienda. Devono anche leggere e scrivere dati condivisi sui quali lavorano tutti gli sviluppatori. Per farlo, Mateo dispone logicamente i dati in aws-s3-bucket utilizzando uno schema di prefisso della chiave Amazon S3 come illustrato nella seguente figura.

```
/aws-s3-bucket
    /home
        /zhang
        /major
    /share
        /developers
        /managers
```

Mateo divide il `/aws-s3-bucket` in un set di directory principali per ogni dipendente e un'area condivisa per gruppi di sviluppatori e responsabili.

A questo punto Mateo crea un set di policy per assegnare le autorizzazioni agli utenti e ai gruppi di utenti:

- Accesso alla directory principale per Zhang: Mateo collega una policy a Wei che gli permette di leggere, scrivere ed elencare tutti gli oggetti con il prefisso della chiave Amazon S3 `/aws-s3-bucket/home/zhang/`
- Accesso alla directory principale per Major: Mateo collega una policy a Mary che le permette di leggere, scrivere ed elencare tutti gli oggetti con il prefisso della chiave Amazon S3 `/aws-s3-bucket/home/major/`
- Accesso alla directory condivisa per il gruppo di utenti Sviluppatori: Mateo collega una policy al gruppo di utenti che consente agli sviluppatori di leggere, scrivere ed elencare qualsiasi oggetto in `/aws-s3-bucket/share/developers/`

- Accesso alla directory condivisa per il gruppo di utenti Responsabili: Mateo collega una policy al gruppo di utenti che consente ai responsabili di leggere, scrivere ed elencare qualsiasi oggetto in `/aws-s3-bucket/share/managers/`

> **Note**
>
> Amazon S3 non fornisce automaticamente l'autorizzazione a un utente che crea un bucket o un oggetto di eseguire altre operazioni su quell'oggetto o bucket. Pertanto, nelle policy IAM è necessario fornire esplicitamente agli utenti l'autorizzazione per utilizzare le risorse Amazon S3 che creano.

Per esempi della probabile struttura di queste policy, consulta Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service. Per informazioni su come le policy vengono valutate in fase di runtime, consulta Logica di valutazione delle policy.

Modifica della funzione lavorativa dell'utente

A un certo punto, uno degli sviluppatori, Zhang Wei, cambia le funzioni lavorative e diventa un responsabile. Si presuppone che non abbia più bisogno di accedere ai documenti nella directory `share/developers`. Mateo, come amministratore, sposta Wei nel gruppo di utenti `Managers` e lo rimuove dal gruppo `Developers`. Con quella semplice riassegnazione, Wei ottiene automaticamente tutte le autorizzazioni concesse al gruppo di utenti `Managers`, ma non è più in grado di accedere a dati nella directory `share/developers`.

Integrazione con un business di terze parti

Le organizzazioni spesso lavorano con aziende partner, consulenti e appaltatori. Example Corp ha un partner che si chiama Widget Company e un dipendente di Widget Company che si chiama Shirley Rodriguez deve inserire dati in un bucket perché siano utilizzati da Example Corp. Nikki crea un gruppo di utenti chiamato WidgetCoe un nome utente `Shirley` e aggiunge Shirley al gruppo di utenti. WidgetCo Nikki crea anche un bucket speciale per Shirley chiamato aws-s3-bucket1.

Nikki aggiorna le policy esistenti o ne aggiunge di nuove per aiutare l'azienda partner Widget Company. Ad esempio, Nikki può creare una nuova politica che nega ai membri del gruppo di WidgetCo utenti la possibilità di utilizzare azioni diverse dalla scrittura. Questa policy è necessaria solo se è presente una policy ampia che offre a tutti gli utenti l'accesso a un'ampia gamma di operazioni Amazon S3.

Tutorial IAM

I seguenti tutorial presentano end-to-end procedure complete per le attività comuni per AWS Identity and Access Management (IAM). Gli scenari presentati sono solo esempi con nomi di società e utenti fittizi destinati a essere usati in un ambiente di laboratorio. Il loro scopo è di fornire linee guida di carattere generico. Non devono essere utilizzati direttamente nell'ambiente di produzione, senza un'accurata opera di revisione e adattamento alle necessità esclusive del tuo ambiente lavorativo.

Tutorial

- Tutorial IAM: Concessione dell'accesso alla console di fatturazione
- Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM
- Tutorial IAM: Creazione e collegamento della prima policy gestita dal cliente
- Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag
- Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA

Tutorial IAM: Concessione dell'accesso alla console di fatturazione

Il Account AWS proprietario (Utente root dell'account AWS) può concedere agli utenti e ai ruoli IAM l'accesso ai AWS Billing and Cost Management dati per loro conto Account AWS. Le istruzioni che seguono consentono di impostare uno scenario pretestato. Questo scenario ti consente di acquisire esperienza pratica nella configurazione delle autorizzazioni di fatturazione senza preoccuparsi di influire sull'account AWS di produzione principale.

Prerequisiti

Prima di eseguire i passaggi in questo tutorial, completa le seguenti operazioni preliminari:

- Crea un test Account AWS.
- Accedi al test Account AWS come utente root.
- Registra il Account AWS numero del tuo account di test in modo da poterlo utilizzare nel tutorial. In questo tutorial utilizziamo il numero di account di esempio 111122223333. Ogni volta che un passaggio utilizza quel numero account, sostituiscilo con il tuo numero di account di prova.

Fase 1: Attiva l'accesso IAM alle informazioni di fatturazione sul tuo account di test AWS

In questo scenario, accedi al test Account AWS come utente root per concedere a IAM l'accesso alle informazioni di fatturazione. Quando concedi a IAM l'accesso alle informazioni di fatturazione, consente agli utenti e ai ruoli IAM di accedere alla AWS Billing and Cost Management console. Questa impostazione non concede agli utenti e ai ruoli IAM le autorizzazioni necessarie per queste pagine della console, ma concede l'accesso agli utenti o ai ruoli IAM che dispongono delle policy IAM richieste. Se le policy sono già associate agli utenti o ai ruoli IAM ma questa impostazione non è abilitata, le autorizzazioni concesse da tali policy non saranno valide.

> **Note**
>
> Account AWS creato utilizzando AWS Organizations avere l'accesso IAM alle informazioni di fatturazione abilitato per impostazione predefinita.

Fase 2: crea utenti e gruppi di prova

In questo scenario, concedi agli utenti IAM l'accesso alla console di fatturazione e crei due utenti:

- Pat Candella

  Pat è membro del dipartimento finanziario e si occupa di fatturazione e pagamenti. Pat richiede l'accesso completo alle informazioni di fatturazione nel tuo. Account AWS

- Terry Whitlock

  Terry fa parte del tuo reparto di assistenza IT. La maggior parte delle volte Terry non richiede l'accesso alla console di fatturazione, ma a volte ne ha bisogno per rispondere alle domande dei dipendenti del settore finanziario.

Fase 3: Creazione di un ruolo per concedere l'accesso alla console AWS Billing

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. Un ruolo IAM è simile a un utente IAM, in quanto è un'identità AWS con policy di autorizzazioni che determinano ciò che l'identità può e non può fare in AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate, come password o chiavi di accesso. Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo. Puoi utilizzare i ruoli per delegare l'accesso a utenti, applicazioni o servizi che normalmente non hanno accesso alle tue risorse. AWS In questo scenario crei un ruolo che Terry Whitlock può assumere per accedere alla console di fatturazione.

[Fase 4: Test dell'accesso alla console]

Dopo aver completato le attività fondamentali sei pronto per testare la policy. I test garantiscono che la policy funzioni nel modo desiderato. Testando l'accesso di ogni utente puoi confrontare le esperienze degli utenti.

Prerequisiti

Prima di eseguire i passaggi in questo tutorial, completa le seguenti operazioni preliminari:

- Crea un test Account AWS.
- Accedi al test Account AWS come utente root.
- Registra il Account AWS numero del tuo account di test in modo da poterlo utilizzare nel tutorial. In questo tutorial utilizziamo il numero di account di esempio 111122223333. Ogni volta che un passaggio utilizza quel numero account, sostituiscilo con il tuo numero di account di prova.

Fase 1: Attiva l'accesso IAM alle informazioni di fatturazione sul tuo account di test AWS

In questo scenario, accedi al test Account AWS come utente root per concedere a IAM l'accesso alle informazioni di fatturazione. Quando concedi l'accesso alle informazioni di fatturazione, consente agli utenti e ai ruoli IAM di accedere alla AWS Billing and Cost Management console. Questa impostazione non concede agli utenti e ai ruoli IAM le autorizzazioni necessarie per queste pagine della console, ma concede l'accesso agli utenti o ai ruoli IAM che dispongono delle policy IAM richieste.

> **Note**
>
> Account AWS creato utilizzando AWS Organizations avere l'accesso IAM alle informazioni di fatturazione abilitato per impostazione predefinita.

Per attivare l'accesso degli utenti e dei ruoli IAM alla console di Gestione costi e fatturazione.

1. Accedi a AWS Management Console con le tue credenziali utente root (in particolare, l'indirizzo email e la password che hai usato per creare il tuo AWS account).

2. Nella barra di navigazione seleziona il tuo nome account, quindi scegli Account.
3. Scorri la pagina verso il basso fino a trovare la sezione Accesso utente e ruolo IAM alle informazioni di fatturazione, quindi seleziona Modifica.
4. Seleziona la casella di controllo Activate IAM Access (Attiva l'accesso IAM) per attivare l'accesso alle pagine della console di Gestione costi e fatturazione.
5. Scegli Update (Aggiorna).

   La pagina mostra il messaggio che l'accesso utente/ruolo IAM alle informazioni di fatturazione è stato attivato.

Nella fase successiva di questo tutorial collegherai le policy IAM per concedere o rifiutare l'accesso a funzionalità di fatturazione specifiche.

Fase 2: crea utenti e gruppi di prova

Il tuo AWS account di test non ha alcuna identità definita ad eccezione dell'utente root. Per fornire l'accesso alle informazioni di fatturazione, vengono create identità aggiuntive a cui si concede l'autorizzazione per accedere ai dati di fatturazione.

Creare utenti e gruppi di prova

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Nel pannello di navigazione seleziona Users (Utenti), quindi seleziona Add users (Aggiungi utenti).

> **Note**
>
> Se hai abilitato IAM Identity Center, AWS Management Console viene visualizzato un promemoria che ti ricorda che è meglio gestire l'accesso degli utenti in IAM Identity Center. In questo tutorial, gli utenti IAM creati impareranno a fornire l'accesso ai dati di fatturazione. Se sono stati creati utenti nel Centro identità IAM, viene assegnato loro il set di autorizzazioni Fatturazione utilizzando il Centro identità IAM anziché IAM.

3. In Nome utente, inserisci **`pcandella`**. I nomi non possono contenere spazi.
4. Seleziona la casella di selezione accanto a Fornisci l'accesso utente a AWS Management Console— opzionale, quindi scegli «Vuoi creare un utente IAM».
5. Per Console password (Password della console), seleziona Autogenerated password (Password generata automaticamente).
6. Deseleziona la casella accanto a L'utente deve creare una nuova password all'accesso successivo (consigliato), quindi seleziona Successivo. Poiché questo utente IAM è a scopo di prova, scaricheremo la password da utilizzare durante la procedura di verifica.
7. Nella pagina Set permissions (Imposta autorizzazioni), in Permissions options (Opzioni di autorizzazione), seleziona Add user to group (Aggiungi utente al gruppo). Quindi, in User groups (Gruppi di utenti), seleziona Create group (Crea gruppo).
8. Nella pagina Create user group (Crea gruppo di utenti), in User group name (Nome gruppo di utenti), inserisci **`BillingGroup`**. Quindi, in Politiche di autorizzazione, seleziona la politica di AWS gestione delle funzioni lavorative e della fatturazione.
9. Seleziona Create user group (Crea gruppo di utenti) per tornare alla pagina Set permissions (Imposta autorizzazioni).
10. In Gruppi di utenti, seleziona la casella di **`BillingGroup`** creato in precedenza.
11. Seleziona Next (Successivo) per passare alla pagina Review and create (Rivedi e crea).
12. Nella pagina Rivedi e crea, consulta l'elenco dei membri del gruppo di utenti da aggiungere al nuovo utente. Una volta pronto per continuare, seleziona Create user (Crea utente).
13. Nella pagina Recupera password, seleziona Scarica il file .csv per salvare un file .csv con le informazioni sulle credenziali dell'utente (URL di connessione, nome utente e password).

    Salva questo file per utilizzarlo come riferimento quando accedi AWS come utente IAM
14. Seleziona Torna all'elenco degli utenti

15. Ripetere questa procedura utilizzando le seguenti modifiche per creare l'utente per Terry Whitlock e un gruppo per gli utenti dell'assistenza.

    a. Nella fase 3, per Nome utente, inserisci **`twhitlock`**.

    b. Nella fase 8, per Nome gruppo di utenti, inserisci **`SupportGroup`**. Quindi, in Politiche di autorizzazione, seleziona la politica della funzione AWS gestita del lavoro. SupportUser

Puoi esaminare i nuovi utenti, gruppi e ruoli IAM negli elenchi della console. Per ogni elemento creato puoi selezionare il nome per visualizzarne i dettagli. Quando visualizzi i dettagli dell'utente, la console mostra la Fatturazione elencata in Politiche di autorizzazione per **`pcandella`** ed SupportUserelencata in Politiche di autorizzazione per. **`twhitlock`**

Per ulteriori informazioni sull'utilizzo delle policy per concedere agli utenti IAM l'accesso alle funzionalità di AWS Billing and Cost Management , consulta la sezione Utilizzo di policy basate sull'identità (policy IAM) per la AWS Billing nella Guida per l'utente di AWS Billing .

Fase 3: Creazione di un ruolo per concedere l'accesso alla console AWS Billing

Puoi utilizzare un ruolo per concedere agli utenti IAM l'accesso alla console di fatturazione. I ruoli forniscono credenziali temporanee che gli utenti possono assumere quando necessario. In questo tutorial, l'utente **`twhitlock`** deve essere in grado di accedere ai dati di fatturazione quando una richiesta di assistenza del dipartimento finanziario necessita di indagare su un problema.

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo Account AWS indirizzo email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Nel riquadro di navigazione, seleziona Utenti, quindi seleziona l'utente **`twhitlock`** per visualizzare i dettagli dell'utente. Copia l'ARN dell'utente **`twhitlock`** negli appunti.

3. Nel pannello di navigazione, seleziona Ruoli, quindi Crea ruolo.
4. Nella pagina Seleziona entità attendibile, seleziona Policy di attendibilità personalizzata, quindi in Modifica istruzione completa i seguenti campi:

   - Aggiungi azioni per STS: verifica che AssumeRolesia selezionato.
   - Aggiungi un principale: seleziona Aggiungi per visualizzare la finestra di dialogo Aggiungi principale. Per Tipo principale, seleziona Utenti IAM, quindi per ARN incolla l'ARN dell'utente twhitlock copiato negli appunti nella fase 16. Quindi, seleziona Aggiungi principale.
5. Seleziona Successivo per andare alla pagina Aggiungi autorizzazioni.
6. In Politiche di autorizzazione nella casella del filtro, inserisci **`Billing`** e quindi seleziona la politica di fatturazione della funzione AWS gestita per le funzioni lavorative.
7. Scegli Avanti due volte per andare alla pagina Nomina, verifica e crea. In Nome ruolo, inserisci **`TempBillingAccess`** e seleziona Crea ruolo.

   Riceverai una notifica che il ruolo è stato creato. Visualizza il ruolo per visualizzarne i dettagli. Nella sezione Riepilogo, prendi nota delle seguenti informazioni:

   - Per impostazione predefinita, Durata massima sessione è 1 ora. Dopo questo periodo, l'utente che ha assunto il ruolo torna alle autorizzazioni dell'account di base. Se l'utente desidera continuare a utilizzare le autorizzazioni dei ruoli, deve cambiare nuovamente ruolo. Puoi modificare il ruolo per aumentare la durata massima. La durata della sessione più lunga possibile è di 12 ore.
   - Collegamento per cambiare ruoli nella console. Puoi copiare il link per fornirlo direttamente agli utenti che aggiungi come principali nella policy di attendibilità. È possibile visualizzare e modificare la policy di attendibilità dalla scheda Relazioni di attendibilità.

Fase 4: Test dell'accesso alla console

Consigliamo di verificare l'accesso autenticandosi come ciascuno degli utenti di prova in modo da poter visualizzare la potenziale esperienza degli utenti. Usare le fasi che seguono per effettuare l'accesso utilizzando entrambi gli account di prova per visualizzare la differenza tra i diritti di accesso.

Per testare gli accessi di fatturazione effettuando l'accesso con entrambi gli utenti di test

1. Utilizza l'ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. AWS

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

2. Autenticarsi con ogni utente utilizzando la procedura descritta qui di seguito in modo da confrontare le diverse esperienze degli utenti.

   Full access (Accesso completo)

   a. Accedi al tuo account Account AWS come utente. **`pcandella`**
   b. Nella barra di navigazione, scegli pcandella@111122223333, quindi scegli Pannello di controllo di fatturazione.
   c. Sfoglia le pagine e seleziona i vari pulsanti per essere certo di disporre delle autorizzazioni di modifica completa.

   Nessun accesso

   a. Accedi al tuo Account AWS account come utente**`twhitlock`**.
   b. Nella barra di navigazione, scegli twhitlock@111122223333, quindi scegli Pannello di controllo di fatturazione.
   c. Viene visualizzato un messaggio che indica che sono necessarie le autorizzazioni. Nessun dato di fatturazione è visibile.

   Cambio del ruolo per aumentare l'accesso

   a. Accedi al tuo Account AWS account come utente**`twhitlock`**.
   b. Nella barra di navigazione, scegli twhitlock@111122223333, quindi scegli Cambia ruolo.

      Viene visualizzata la pagina Cambia ruolo. Immetti le informazioni come segue:

      - Account: 111122223333
      - Ruolo-**`TempBillingAccess`**

Seleziona Cambia ruolo

In alternativa, puoi utilizzare l'URL fornito in Collegamento per cambiare ruoli nella console per aprire la pagina Cambia ruolo.

c. La console visualizza la AWS Billing dashboard e la barra di navigazione mostra TempBillingAccess@111122223333.

Riepilogo

Ora hai completato i passaggi necessari per fornire agli utenti IAM l'accesso alla console AWS Billing . Di conseguenza, hai visto in prima persona com'è l'esperienza della console di fatturazione degli utenti. È ora possibile procedere all'implementazione di questa logica nell'ambiente di produzione a proprio piacimento.

Risorse correlate

Per informazioni correlate disponibili nella Guida per l'utente di AWS Billing , consulta le risorse seguenti:

- Attivazione dell'accesso alla console AWS Billing
- AWS Esempi di politiche di fatturazione
- Utilizzo di politiche basate sull'identità (politiche IAM) per la fatturazione AWS
- Migrazione del controllo degli accessi per AWS Billing

Per informazioni correlate nella Guida per l'utente di IAM, consulta le risorse seguenti:

- Policy gestite e policy inline
- Controllo dell'accesso utente IAM alla AWS Management Console
- Collegamento di una policy a un gruppo di utenti IAM

Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM

In questo tutorial viene descritto come utilizzare un ruolo per delegare l'accesso a risorse che si trovano in diversi Account AWS di tua proprietà denominati Produzione e Sviluppo. Puoi condividere

le risorse di un account con gli utenti di un altro account. Configurando in questo modo l'accesso fra account, non dovrai creare singoli utenti IAM per ogni account. Inoltre, gli utenti non devono uscire da un account e accedere a un altro per utilizzare risorse in Account AWS diversi. Dopo aver configurato il ruolo, vedrai come utilizzarlo tramite AWS Management Console AWS CLI, e l'API.

> **Note**
>
> I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, si supponga di disporre di un account nella regione Stati Uniti occidentali (California settentrionale) nella partizione `aws` standard. Hai anche un account nella regione Cina (Pechino) nella partizione `aws-cn`. Non è possibile utilizzare una policy basata sulle risorse Amazon S3 nel tuo account nella regione Cina (Pechino) per consentire l'accesso agli utenti del tuo account `aws` standard.

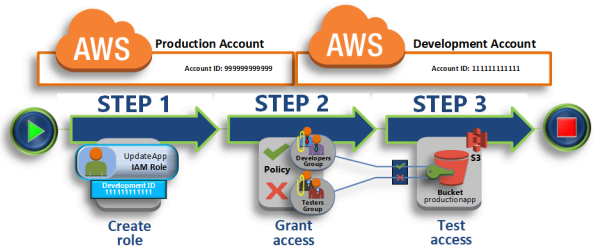
In questa esercitazione, l'account Produzione gestisce le applicazioni live. Sviluppatori e tester utilizzano l'account Sviluppo come sanbox per testare liberamente le applicazioni. In ciascun account puoi archiviare le informazioni sull'applicazione in bucket Amazon S3. Gli utenti IAM vengono gestiti nell'account Sviluppo in cui sono presenti due gruppi di utenti IAM: Developer e Tester. Gli utenti di entrambi i gruppi di utenti dispongono di autorizzazioni per lavorare nell'account Sviluppo e accedere alle sue risorse. Occasionalmente, uno sviluppatore deve aggiornare le applicazioni attive nell'account Produzione. Gli sviluppatori archiviano queste applicazioni in un bucket Amazon S3 denominato `productionapp`.

Alla fine di questo tutorial, si dispone di quanto segue:

- Utenti nell'account Sviluppo (account che concede fiducia) che possono assumere un ruolo specifico nell'account Produzione.
- Un ruolo nell'account Produzione (account attendibile) che può accedere a uno specifico bucket Amazon S3.
- Il bucket `productionapp` creato nell'account Produzione.

Gli sviluppatori possono utilizzare il ruolo in AWS Management Console per accedere al `productionapp` bucket nell'account Production. Inoltre, possono accedere al bucket utilizzando chiamate API autenticate tramite credenziali provvisorie fornite dal ruolo. La stessa operazione eseguita da un tester avrà esito negativo.

Questo flusso di lavoro ha tre fasi di base:



Creazione di un ruolo nell'account Produzione

Innanzitutto, si utilizza il AWS Management Console per stabilire un rapporto di fiducia tra l'account di produzione (numero ID assistere a 9999) e l'account di sviluppo (numero ID 1111). Si inizia creando un ruolo IAM denominato. UpdateApp Quando crei il ruolo, devi definire l'account Sviluppo come entità attendibile e specificare una policy di autorizzazioni che consenta agli utenti attendibili di aggiornare il bucket `productionapp`.

Concedi autorizzazione per l'accesso al ruolo

In questa sezione, puoi modificare la policy del gruppo di utenti IAM per negare l'accesso al ruolo `UpdateApp` ai Tester. Perché i tester hanno PowerUser accesso in questo scenario e devi negare esplicitamente la possibilità di utilizzare il ruolo.

Accesso al test tramite cambio di ruoli

Infine, in qualità di Sviluppatore utilizzerai il ruolo `UpdateApp` per aggiornare il bucket `productionapp` nell'account Produzione. Scopri come accedere al ruolo tramite la AWS console, l'e l' AWS CLI API.

Prerequisiti

Questo tutorial presuppone che tu abbia a disposizione quanto segue:

- È possibile utilizzarne due Account AWS distinti, uno per rappresentare l'account Development e uno per rappresentare l'account Production.
- Utenti e gruppi di utenti nell'account Sviluppo, creati e configurati in questo modo:

Utente	Gruppo di utenti	Autorizzazioni
David	Sviluppatori	Entrambi gli utenti possono accedere e utilizzare l' AWS Management Console account Development.

Utente	Gruppo di utenti	Autorizzazioni
Jane	Tester	

- Non è necessario creare utenti o gruppi di utenti nell'account Produzione.
- Un bucket Amazon S3 creato nell'account Produzione. Nel tutorial puoi chiamarlo `ProductionApp`, ma dato che i nomi dei bucket S3 devono essere univoci a livello globale, dovrai selezionare un nome diverso.

Creazione di un ruolo nell'account Produzione

Puoi consentire agli utenti di uno Account AWS di accedere alle risorse di un altro Account AWS. A tale scopo, creare un ruolo che definisca chi può accedervi e quali autorizzazioni concedere agli utenti che lo assumono.

In questo passaggio del tutorial, dovrai creare il ruolo nell'account Produzione e impostare l'account Sviluppo come entità attendibile. Inoltre, dovrai limitare le autorizzazioni del ruolo al solo accesso di lettura e scrittura per il bucket `productionapp`. Chiunque abbia ricevuto l'autorizzazione a usare il ruolo potrà leggere e scrivere nel bucket `productionapp`.

Prima di poter creare un ruolo, è necessario l'ID dell'account Development Account AWS. A ognuno Account AWS è assegnato un identificatore ID account univoco.

Per ottenere l'ID di sviluppo Account AWS

1. Accedi AWS Management Console come amministratore dell'account Development e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nella barra di navigazione, selezionare Support (Supporto) e Support Center (Centro supporto). Il numero di account a 12 cifre (ID) correntemente collegato viene visualizzato nel pannello di navigazione Centro assistenza. Per questo scenario, puoi utilizzare l'ID account 111111111111 per l'account Sviluppo. Tuttavia, devi utilizzare un ID account valido se stai usando questo scenario nell'ambiente di test.

Creare un ruolo nell'account di produzione che possa essere utilizzato dall'account di sviluppo

1. Accedi AWS Management Console come amministratore dell'account di produzione e apri la console IAM.

2. Prima di creare il ruolo, prepara la policy gestita che definisce le autorizzazioni per i requisiti del ruolo. La policy verrà collegata al ruolo in una fase successiva.

   Impostare l'accesso in lettura e scrittura al bucket `productionapp`. Sebbene AWS fornisca alcune policy gestite di Amazon S3, non ce n'è una che fornisca l'accesso in lettura e scrittura a un singolo bucket Amazon S3. Se lo desideri, puoi creare una policy personalizzata.

   Nel pannello di navigazione, seleziona Policy e Crea policy.

3. Seleziona la scheda JSON e copia il testo dal documento della seguente policy JSON. Incollare il testo nella casella di testo JSON, sostituendo l'ARN della risorsa (`arn:aws:s3:::productionapp`) con quello per reale per il bucket Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
       ],
      "Resource": "arn:aws:s3:::productionapp"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::productionapp/*"
    }
  ]
}
```

L'azione `ListAllMyBuckets` concede l'autorizzazione per elencare tutti i bucket di proprietà del mittente autenticato della richiesta. L'autorizzazione `ListBucket` consente agli utenti di visualizzare gli oggetti del bucket `productionapp`. Le autorizzazioni `GetObject`, `PutObject` e `DeleteObject` consentono agli utenti di visualizzare, aggiornare ed eliminare i contenuti del bucket `productionapp`.

4. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

5. Nella pagina Verifica policy, digita **`read-write-app-bucket`** come nome della policy. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy viene inserita nell'elenco delle policy gestite.

6. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.
7. Scegli il tipo di ruolo Un Account AWS.
8. In Account ID (ID account), digitare l'ID dell'account Sviluppo.

   Questo tutorial utilizza come esempio l'ID account **111111111111** per l'account di sviluppo. Tuttavia, è consigliabile utilizzare un ID account valido. Se utilizzi un ID account non valido, come ad esempio **111111111111**, IAM non ti consentirà di creare il nuovo ruolo.

   Per il momento non è necessario richiedere un ID esterno, né chiedere agli utenti un'autenticazione a più fattori (MFA) per assumere il ruolo. Tali opzioni possono rimanere deselezionate. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

9. Selezionare Next:Permissions (Avanti:Autorizzazioni) per impostare le autorizzazioni associate al ruolo.
10. Seleziona la casella accanto alla policy creata in precedenza.

> **Suggerimento**
>
> In Filter (Filtro) selezionare Customer managed (Gestite dal cliente) per visualizzare solo le policy create. Il filtro nasconde le policy create da AWS per semplificare la ricerca.

Quindi, seleziona Next (Successivo).

11. (Facoltativo) Aggiungi metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
12. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
13. Dopo avere rivisto il ruolo, fare clic su Create Role (Crea ruolo).

    Il ruolo `UpdateApp` viene visualizzato nell'elenco dei ruoli.

A questo punto, è necessario ottenere l'Amazon Resource Name (ARN), un identificatore univoco per il ruolo. Quando modifichi le policy dei gruppi Developer e Tester, specificare l'ARN del ruolo per concedere o negare le autorizzazioni.

Per ottenere l'ARN per UpdateApp

1. Nel pannello di navigazione della console IAM seleziona Ruoli.
2. Nell'elenco dei ruoli, selezionare il ruolo `UpdateApp`.
3. Nella sezione Summary (Riepilogo) del riquadro dei dettagli, copiare il valore Role ARN (ARN ruolo).

   L'ID dell'account di produzione è 999999999999, pertanto l'ARN del ruolo sarà `arn:aws:iam::999999999999:role/UpdateApp`. Assicurati di fornire l' Account AWS ID reale dell'account di produzione.

A questo punto, è stato stabilita una relazione di trust tra gli account di Produzione e Sviluppo. È stato possibile creare un ruolo nell'account Produzione che identifica l'account Sviluppo come entità attendibile. Inoltre, hai definito le operazioni consentite agli utenti che passano al ruolo `UpdateApp`.

Quindi, bisogna modificare le autorizzazioni per i gruppi di utenti.

Concedi autorizzazione per l'accesso al ruolo

A questo punto, i membri dei gruppi Developer e Tester dispongono delle autorizzazioni per testare liberamente le applicazioni dell'account Sviluppo. Usare i seguenti passaggi necessari per aggiungere le autorizzazioni per il cambio di ruolo.

Modificare il gruppo di utenti Developers per consentire loro di passare al UpdateApp ruolo

1. Accedi come amministratore nell'account Sviluppo e apri la console IAM.
2. Seleziona Gruppi di utenti e scegli Developer.
3. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi Crea policy in linea.
4. Scegli la scheda JSON.
5. Aggiungere la seguente istruzione di policy per consentire l'azione `AssumeRole` sul ruolo `UpdateApp` nell'account Produzione. Assicurati di modificare *`PRODUCTION-ACCOUNT-ID`* nell'`Resource`elemento con l' Account AWS ID effettivo dell'account di produzione.

   ```
   {
     "Version": "2012-10-17",
     "Statement": {
       "Effect": "Allow",
       "Action": "sts:AssumeRole",
       "Resource": "arn:aws:iam::PRODUCTION-ACCOUNT-ID:role/UpdateApp"
     }
   }
   ```

   L'effetto `Allow` consente in modo esplicito al gruppo Developer l'accesso al ruolo `UpdateApp` dell'account Produzione. Qualsiasi sviluppatore potrà accedere al ruolo.
6. Scegli Verifica policy.
7. Digita un nome, ad esempio **`allow-assume-S3-role-in-production`**.
8. Scegli Crea policy.

Nella maggior parte degli ambienti, la procedura seguente risulta superflua. Se, tuttavia, utilizzi PowerUserAccess le autorizzazioni, alcuni gruppi potrebbero già essere in grado di cambiare ruolo. La procedura descritta di seguito mostra come aggiungere un'autorizzazione `"Deny"` al gruppo Tester, per impedire ai suoi membri di assumere il ruolo. Se questa procedura non è necessaria nel tuo ambiente, è preferibile non aggiungerla. Le autorizzazioni `"Deny"` di tipo generale sono

più complicate da gestire e comprendere. Utilizza le autorizzazioni "Deny" solo quando non sono disponibili alternative migliori.

Come modificare il gruppo di utenti Tester e negare l'autorizzazione ad assumere il ruolo **UpdateApp**

1. Seleziona Gruppi di utenti quindi Tester.
2. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi Crea policy in linea.
3. Scegli la scheda JSON.
4. Aggiungere la seguente istruzione di policy per negare l'operazione AssumeRole nel ruolo UpdateApp. Assicurati di modificare *PRODUCTION-ACCOUNT-ID* nell'Resourceelemento con l' Account AWS ID effettivo dell'account di produzione.

   ```
   {
     "Version": "2012-10-17",
     "Statement": {
       "Effect": "Deny",
       "Action": "sts:AssumeRole",
       "Resource": "arn:aws:iam::PRODUCTION-ACCOUNT-ID:role/UpdateApp"
     }
   }
   ```

   L'effetto Deny impedisce in modo esplicito al gruppo Tester l'accesso al ruolo UpdateApp dell'account Produzione. Se un tester cerca di accedere al ruolo riceve un messaggio di accesso negato.
5. Scegli Verifica policy.
6. Digita un nome come **deny-assume-S3-role-in-production**.
7. Scegli Crea policy.

Il gruppo Developer ora dispone delle autorizzazioni per utilizzare il ruolo UpdateApp nell'account di produzione. Il gruppo di utenti Tester invece non può utilizzare il ruolo UpdateApp.

Successivamente, vedrai come David, uno sviluppatore, può accedere al bucket productionapp nell'account Produzione. David può accedere al bucket dal AWS Management Console, dall'o dall' AWS CLI API. AWS

Accesso al test tramite cambio di ruoli

Al termine dei primi due passaggi del tutorial, disponi di un ruolo che concede l'accesso a una risorsa dell'account Produzione. Hai anche creato un gruppo di utenti nell'account Sviluppo con utenti autorizzati a utilizzare tale ruolo. In questo passaggio viene illustrato come testare il passaggio a quel ruolo tramite AWS Management Console AWS CLI, l'e l' AWS API.

> **Important**
>
> È possibile passare a un ruolo solo dopo avere effettuato l'accesso come utente IAM o utente federato. Inoltre, se avvii un'istanza Amazon EC2 per eseguire un'applicazione, questa può assumere un ruolo tramite il proprio profilo dell'istanza. Non è possibile passare a un ruolo se l'accesso è stato effettuato come Utente root dell'account AWS.

Cambio di ruoli (Console)

Se David ha bisogno di lavorare nell'ambiente di produzione di AWS Management Console, può farlo utilizzando Switch Role. Specificando l'ID account o l'alias e il nome del ruolo, le sue autorizzazioni passano immediatamente a quelle consentite dal ruolo. Potrà quindi utilizzare la console per lavorare con il bucket `productionapp`, ma non sarà in grado utilizzare le altre risorse dell'account Produzione. Inoltre, mentre utilizza il ruolo, David non può sfruttare i suoi privilegi di utente avanzato, validi per l'account Sviluppo, perché non si possono attivare più set di autorizzazioni contemporaneamente.

> **Important**
>
> Il cambio di ruolo utilizzando il funziona AWS Management Console solo con account che non richiedono un`ExternalId`. Ad esempio, si supponga di concedere l'accesso al proprio account a terzi e di richiedere un `ExternalId` in un elemento `Condition` nella policy di autorizzazione. In tal caso, la terza parte può accedere all'account solo utilizzando l' AWS API o uno strumento da riga di comando. Le terze parti non possono utilizzare la console perché non sono in grado di fornire un valore per `ExternalId`. Per ulteriori informazioni su questo scenarioCome utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi, consulta e Come abilitare l'accesso tra più account AWS Management Console al blog sulla AWS sicurezza.

IAM fornisce due alternative per accedere alla pagina Switch Role (Cambia ruolo):

- David riceve dal suo amministratore un link che rimanda a una configurazione "Switch Role" (Cambia ruolo) predefinita. Il collegamento viene fornito all'amministratore nella pagina finale della procedura guidata Create role (Crea ruolo) oppure nella pagina Role Summary (Riepilogo ruolo) per un ruolo tra più account. Selezionando questo link, David viene indirizzato alla pagina Switch Role (Cambia ruolo) con i campi Account ID (ID account) e Role name (Nome ruolo) già compilati. David non deve fare altro che selezionare Switch Roles (Cambia ruolo).
- Anziché spedire un'e-mail con il link, l'amministratore invia il numero Account ID (ID account) e i valori per Role Name (Nome ruolo). Per cambiare ruolo, David deveinserire manualmente i valori. Tale procedura viene descritta di seguito.

Come assumere un ruolo

1. David accede AWS Management Console utilizzando il suo utente normale nel gruppo di utenti Development.
2. Seleziona il link che l'amministratore gli ha inviato per e-mail. A questo punto, David viene indirizzato alla pagina Switch Role (Cambia ruolo) che contiene già le informazioni relative all'ID account o all'alias e il nome del ruolo.

   oppure

   David seleziona il proprio nome nel menu Identity (Identità) della barra di navigazione, quindi sceglie Switch Roles (Cambia ruolo).

   Se questa è la prima volta che David cerca di accedere alla pagina Switch Role (Cambia ruolo) in questo modo, verrà visualizzata una pagina introduttiva di Switch Role (Cambia ruolo) In questa pagina sono riportate ulteriori informazioni su come il cambio di ruolo può consentire agli utenti di gestire le risorse su più Account AWS. In questa pagina, David deve selezionare Switch Role (Cambia ruolo) per completare il resto della procedura.
3. Successivamente, per accedere al ruolo, David deve digitare manualmente il numero di ID dell'account Produzione (`999999999999`) e il nome del ruolo (`UpdateApp`).

   Inoltre, David vuole monitorare quali ruoli e autorizzazioni associate sono attualmente attivi su IAM. Per tenere traccia di queste informazioni, digita `PRODUCTION` nella casella di testo Nome di visualizzazione, seleziona l'opzione di colore rosso e quindi seleziona Cambia ruolo.

4. Ora David può utilizzare la console Amazon S3 per lavorare con il bucket Amazon S3 o con qualsiasi altra risorsa per la quale il ruolo `UpdateApp` dispone di autorizzazioni.
5. Al termine, David può tornare alle sue autorizzazioni originali. A tale scopo, seleziona il nome del ruolo PRODUCTION (Produzione) nella barra di navigazione, quindi seleziona Back to David @ 111111111111 (Torna a David @ 111111111111.
6. Se dovesse avere nuovamente bisogno di cambiare ruolo, David potrà selezionare il menu Identity (Identità) nel riquadro di navigazione e troverà già presente la voce PRODUCTION (PRODUZIONE). Non dovrà fare altro che selezionare tale voce per cambiare immediatamente ruolo, senza immettere nuovamente l'account ID e il nome del ruolo.

Cambio di ruoli (AWS CLI)

Se David dovesse avere bisogno di lavorare nell'ambiente Produzione, alla riga di comando, può farlo tramite la AWS CLI. Esegue il comando `aws sts assume-role` e trasferisce l'ARN del ruolo per ottenere le credenziali di sicurezza provvisorie per quel ruolo. Quindi configura tali credenziali nelle variabili di ambiente in modo che AWS CLI i comandi successivi funzionino utilizzando le autorizzazioni del ruolo. Mentre utilizza il ruolo, David non può sfruttare i suoi privilegi di utente avanzato, validi per l'account Sviluppo, perché non si possono attivare più set di autorizzazioni contemporaneamente.

Tutte le chiavi di accesso e i token sono solo esempi e non possono essere utilizzati come mostrato. Sostituiscili con i valori appropriati del tuo ambiente reale.

Come assumere un ruolo

1. David apre una finestra del prompt dei comandi e conferma che il AWS CLI client funziona eseguendo il comando:

```
aws help
```

> **Note**
>
> L'ambiente predefinito di David utilizza le credenziali utente `David` ottenute dal profilo predefinito creato con il comando `aws configure`. Per ulteriori informazioni, consulta Configurazione della AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface .

2. Inizia il processo di cambio ruolo eseguendo il seguente comando per passare al ruolo UpdateApp dell'account Produzione. Ha ricevuto l'ARN del ruolo dall'amministratore che ha creato il ruolo. Il comando richiede anche un nome di sessione (qualsiasi testo è valido).

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/UpdateApp" --role-
session-name "David-ProdUpdate"
```

A questo punto David potrà consultare quanto segue:

```
{
    "Credentials": {
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AQoDYXdzEGcaEXAMPLE2gsYULo
+Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLE
CvSRyh0FW7jEXAMPLEW+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/
uZEXAMPLECihzFB5lTYLto9dyBgSDy
EXAMPLE9/
g7QRUhZp4bqbEXAMPLENwGPyOj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3Uuysg
sKdEXAMPLE1TVastU1A0SKFEXAMPLEiywCC/Cs8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP
+4eZScEXAMPLEsnf87e
NhyDHq6ikBQ==",
        "Expiration": "2014-12-11T23:08:07Z",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

3. David può trovare le tre parti di cui ha bisogno nella sezione Credentials (Credenziali) dell'output.

   - `AccessKeyId`
   - `SecretAccessKey`
   - `SessionToken`

   David deve configurare l' AWS CLI ambiente per utilizzare questi parametri nelle chiamate successive. Per informazioni sui vari modi di configurare le credenziali, consulta Configurare AWS Command Line Interface. Non può utilizzare il comando `aws configure`, perché non supporta l'acquisizione il token della sessione. Tuttavia, può inserire manualmente le informazioni in un file di configurazione. Poiché si tratta di credenziali provvisorie, con una durata relativamente breve, è più facile per aggiungerle all'ambiente della sessione corrente della riga di comando.

4. Per aggiungere i tre valori all'ambiente, David taglia e incolla l'output del passaggio precedente nei comandi seguenti. Per risolvere i problemi con gli accapo presenti nel token della sessione, è possibile tagliare e incollare in un semplice editor di testo. Il testo deve essere inserito come un'unica stringa lunga, anche se qui viene riportato spezzato, per motivi di chiarezza.

   > **Note**
   >
   > L'esempio seguente mostra i comandi forniti nell'ambiente Windows, dove "set" è il comando per creare una variabile di ambiente. Su un computer Linux o MacOS, bisogna utilizzare invece il comando "export". Tutte le altre parti dell'esempio sono valide per tutti i tre ambienti.
   > Per dettagli sull'utilizzo di Tools for Windows Powershell, consulta Passaggio a un ruolo IAM (Tools for Windows PowerShell)

   ```
   set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
   set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   set AWS_SESSION_TOKEN=AQoDYXdzEGcaEXAMPLE2gsYULo
   +Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLECvS
   Ryh0FW7jEXAMPLEW+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/
   uZEXAMPLECihzFB5lTYLto9dyBgSDyEXA
   MPLEKEY9/
   g7QRUhZp4bqbEXAMPLENwGPyOj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3UusKd
   EXAMPLE1TVastU1A0SKFEXAMPLEiywCC/Cs8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP
   +4eZScEXAMPLENhykxiHen
   DHq6ikBQ==
   ```

   A questo punto, tutti i comandi successivi vengono eseguiti con le autorizzazioni del ruolo identificato da tali credenziali (nel caso di David, il ruolo `UpdateApp`).

5. Eseguire il comando per accedere alle risorse dell'account Produzione. In questo esempio, David si limita a elencare il contenuto del suo bucket S3 con il comando seguente.

   ```
   aws s3 ls s3://productionapp
   ```

   Poiché i nomi del bucket Amazon S3 sono universalmente univoci, non è necessario specificare quale ID account possiede il bucket. Per accedere alle risorse di altri AWS servizi, consulta la AWS CLI documentazione del servizio per conoscere i comandi e la sintassi necessari per fare riferimento alle relative risorse.

Utilizzo di AssumeRole (AWS API)

Per aggiornare l'account Produzione da codice, David effettua una chiamata `AssumeRole` per assumere il ruolo `UpdateApp`. La chiamata restituisce le credenziali provvisorie, utilizzabili per accedere al bucket `productionapp` dell'account Produzione. David può utilizzare tali credenziali per effettuare chiamate API per aggiornare il bucket `productionapp`. Tuttavia, non sarà in grado di effettuare chiamate API per accedere alle altre risorse dell'account Produzione, anche se dispone di autorizzazioni da utente avanzato per l'account Sviluppo.

Come assumere un ruolo

1. David richiama `AssumeRole` come parte di un'applicazione Deve specificare l'ARN di `UpdateApp`: `arn:aws:iam::999999999999:role/UpdateApp`.

   La risposta alla chiamata `AssumeRole` include le credenziali temporanee con un `AccessKeyId` e un `SecretAccessKey`. Include anche un'ora `Expiration` che indica quando le credenziali scadono e sarà necessario richiederne di nuove.
2. David utilizza le credenziali provvisorie per inviare una chiamata `s3:PutObject` per aggiornare il bucket `productionapp`. Trasferisce le credenziali alla chiamata API come parametro `AuthParams`. Dato che le credenziali provvisorie del ruolo forniscono solo un accesso in lettura e scrittura al bucket `productionapp`, tutte le altre azioni nell'account Produzione sono negate.

Per un esempio di codice (con Python), consultare Passaggio a un ruolo IAM (AWS API).

Risorse correlate

- Per ulteriori informazioni su utenti e gruppi IAM, consulta Identità IAM (utenti, gruppi di utenti e ruoli).
- Per ulteriori informazioni sui bucket Amazon S3, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.
- Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Riepilogo

Hai completato il tutorial per l'accesso alle API di più account. Hai creato un ruolo per stabilire la relazione di trust con un altro account e hai definito le operazioni che possono essere eseguite dalle

entità affidabili. Successivamente, hai modificato una policy di gruppo per controllare quali utenti IAM possono accedere al ruolo. Come risultato, i developer dell'account Sviluppo possono aggiornare il bucket `productionapp` dell'account Produzione, utilizzando credenziali provvisorie.

Tutorial IAM: Creazione e collegamento della prima policy gestita dal cliente

In questo tutorial, utilizzerai la AWS Management Console per creare una policy gestita dai clienti e poi allegherai tale policy a un utente IAM del tuo Account AWS. La policy che crei consente a un utente di test IAM di accedere direttamente a AWS Management Console con autorizzazioni di sola lettura.

Questo flusso di lavoro ha tre fasi di base:

Fase 1: creazione della policy

Per impostazione predefinita, gli utenti IAM non hanno autorizzazioni per alcuna operazione. Non possono accedere alla Console di gestione AWS né gestire i dati al suo interno, a meno che non venga loro permesso di farlo. In questa fase crei una policy gestita dal cliente che permette agli utenti collegati di accedere alla console.

Fase 2: collegamento della policy

Quando colleghi una policy a un utente, l'utente eredita tutte le autorizzazioni di accesso associate alla policy. In questa fase, colleghi la nuova policy a un utente di test.

Fase 3: test dell'accesso utente

Una volta che la policy è collegata, puoi effettuare l'accesso come utente e testare la policy.

Prerequisiti

Per eseguire le fasi in questo tutorial, devi disporre di quanto segue:

- E a Account AWS cui puoi accedere come utente IAM con autorizzazioni amministrative.
- Un utente IAM di test che non dispone di autorizzazioni assegnate o di appartenenze ai gruppi, come illustrato di seguito:

Nome utente	Group (Gruppo)	Autorizzazioni
PolicyUser	<nessuno>	<nessuno>

Fase 1: creazione della policy

In questo passaggio, crei una policy gestita dai clienti che consente a qualsiasi utente collegato di accedere a AWS Management Console con accesso in sola lettura ai dati IAM.

Per creare la policy per l'utente di test

1. Accedi alla console IAM all'indirizzo https://console.aws.amazon.com/iam/ come utente con autorizzazioni da amministratore.
2. Nel pannello di navigazione, seleziona Policies (Policy).
3. Nel riquadro del contenuto seleziona Create policy (Crea policy).
4. Seleziona l'opzione JSON e copia il testo dal seguente documento della policy JSON. Incolla il testo nella casella di testo JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [ {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateCredentialReport",
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    } ]
}
```

5. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o se si seleziona Rivedi policy nella scheda Editor

visivo, IAM potrebbe ristrutturare la policy per ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

6. Nella pagina Verifica policy, digita **`UsersReadOnlyAccessToIAMConsole`** come nome della policy. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

Fase 2: collegamento della policy

Collega quindi la policy appena creata all'utente di test IAM.

Per collegare la policy all'utente di test

1. Nel pannello di navigazione della console IAM seleziona Policy.
2. Nella parte superiore dell'elenco delle policy, nella casella di ricerca, inizia a digitare **`UsersReadOnlyAccesstoIAMConsole`** finché non viene visualizzata la policy. Quindi scegli il pulsante di opzione accanto a UsersReadOnlyAccessToIAMConsole nell'elenco.
3. Fai clic sul pulsante Actions (Operazioni), quindi scegli Attach (Collega).
4. In Entità IAM scegli l'opzione per filtrare in base a Utenti.
5. Nella casella di ricerca, inizia a digitare **`PolicyUser`** fino a quando l'utente non è visibile nell'elenco. Quindi seleziona la casella accanto a tale utente nell'elenco.
6. Scegli Collega policy.

La policy è stata collegata all'utente di test IAM quindi ora l'utente dispone di accesso in sola lettura alla console IAM.

Fase 3: test dell'accesso utente

Per questa esercitazione, consigliamo di verificare l'accesso autenticandosi come utente di prova in modo da poter visualizzare la potenziale esperienza degli utenti.

Per testare l'accesso accedendo con l'utente di test

1. Accedi alla console IAM all'indirizzo https://console.aws.amazon.com/iam/ con l'utente di prova `PolicyUser`.

2. Esplora le pagine della console e prova a creare un nuovo utente o gruppo. Tieni presente che `PolicyUser` può visualizzare i dati, ma non può creare o modificare i dati IAM esistenti.

Risorse correlate

Per informazioni correlate, consulta le seguenti risorse:

- Policy gestite e policy inline
- Controllo dell'accesso utente IAM alla AWS Management Console

Riepilogo

Hai completato tutte le fasi necessarie per creare e collegare una policy gestita dal cliente. Di conseguenza, è possibile accedere alla console IAM con il proprio account di test per verificare l'esperienza degli utenti.

Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti o ruoli) e alle AWS risorse. È possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni ai principali sulla base dei relativi tag. Quando usi i tag per controllare l'accesso alle tue AWS risorse, consenti ai team e alle risorse di crescere con meno modifiche alle AWS policy. Le politiche ABAC sono più flessibili delle AWS politiche tradizionali, che richiedono di elencare ogni singola risorsa. Per ulteriori informazioni su ABAC e i suoi vantaggi rispetto alle policy tradizionali, consulta A cosa serve ABAC? AWS.

> **Note**
>
> È necessario passare un singolo valore per ogni tag di sessione. AWS Security Token Service non supporta tag di sessione multivalore.

Argomenti

- Panoramica del tutorial

Panoramica del tutorial

Questo tutorial mostra come creare e testare una policy che consente ai ruoli IAM con tag del principale di accedere alle risorse con i tag corrispondenti. Quando un principale effettua una richiesta ad AWS, le autorizzazioni vengono concesse in base al fatto che i tag del principale e quelli della risorsa corrispondano. Questa strategia consente alle persone di visualizzare o modificare solo AWS le risorse necessarie per il proprio lavoro.

Scenario

Si supponga di essere uno sviluppatore responsabile in una grande azienda denominata Example Corporation e di essere un amministratore IAM esperto. Si ha familiarità con la creazione e la gestione di utenti, ruoli e policy IAM. Si desidera garantire che i gli ingegneri dedicati allo sviluppo e i membri del team di garanzia della qualità possano accedere alle risorse di cui hanno bisogno. C'è anche bisogno di una strategia in grado di ridimensionarsi man mano che l'azienda cresce.

Scegli di utilizzare i tag AWS delle risorse e i tag principali dei ruoli IAM per implementare una strategia ABAC per i servizi che la supportano, a cominciare AWS Secrets Manager da. Per informazioni su quali servizi supportano l'autorizzazione basata sui tag, consulta AWS servizi che funzionano con IAM. Per scoprire quali chiavi di condizione di etichettatura puoi utilizzare in una policy con le azioni e le risorse di ciascun servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi. È possibile configurare il provider di identità basato su SAML o web per passare i tag di sessione ad AWS. Quando i dipendenti si uniscono AWS, i loro attributi vengono applicati al responsabile risultante. AWSÈ quindi possibile utilizzare ABAC per consentire o negare le

autorizzazioni sulla base di tali attributi. Per informazioni su come l'utilizzo di tag di sessione con un'identità federata SAML differisce da questa esercitazione, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

I membri dei team Engineering e Quality Assurance fanno parte del progetto Pegasus o Unicorn . È possibile scegliere i seguenti valori di tag lunghi 3 caratteri per progetto e team:

- `access-project` = `peg` per il progetto Pegasus
- `access-project` = `uni` per il progetto Unicorn
- `access-team` = `eng` per il team di Engineering
- `access-team` = `qas` per il team di Quality Assurance

Inoltre, scegli di richiedere il tag di allocazione dei `cost-center` costi per abilitare i report di fatturazione personalizzati AWS . Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing and Cost Management .

Riepilogo delle scelte principali

- I dipendenti eseguono l'accesso con le credenziali dell'utente IAM e quindi assumono il ruolo IAM associato ai relativi team e il progetto. Se l'azienda dispone di un proprio sistema di identità, puoi configurare la federazione per consentire ai dipendenti di assumere un ruolo senza dover passare attraverso gli utenti IAM. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- La stessa policy è collegata a tutti i ruoli. Le operazioni sono consentite o negate in base ai tag.
- I dipendenti possono creare nuove risorse, ma solo se collegano alla risorsa gli stessi tag che sono applicati al loro ruolo. In questo modo i dipendenti possono visualizzare la risorsa dopo averla creata. Gli amministratori non sono più tenuti ad aggiornare le policy con l'ARN delle nuove risorse.
- I dipendenti possono leggere le risorse di proprietà del loro team, indipendentemente dal progetto.
- I dipendenti possono aggiornare ed eliminare le risorse di proprietà del proprio team e progetto.
- Gli amministratori IAM possono aggiungere un nuovo ruolo per i nuovi progetti. Possono creare e associare tag a un nuovo utente IAM per consentire l'accesso al ruolo appropriato. Gli amministratori non sono tenuti a modificare una policy per supportare un nuovo progetto o membro del team.

In questa esercitazione, verranno associati tag a tutte le risorse e ai ruoli del progetto e aggiunte policy ai ruoli per consentire il comportamento precedentemente descritto. La policy risultante

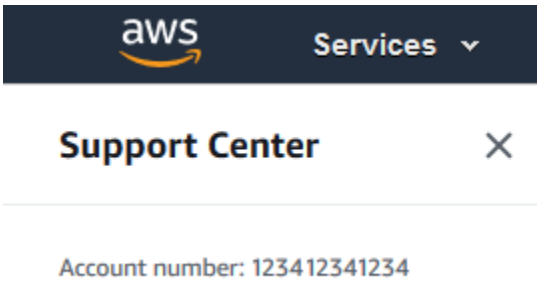
consente ai ruoli `Create`, `Read`, `Update` e `Delete` l'accesso alle risorse contrassegnate con gli stessi tag di progetto e team. La policy consente inoltre l'accesso tra progetti in modalità `Read` per le risorse contrassegnate con lo stesso team.

Prerequisiti

Per eseguire queste fasi in questo tutorial, è necessario quanto segue:

- E a Account AWS cui puoi accedere come utente con autorizzazioni amministrative.
- L'ID account a 12 cifre, che si utilizza per creare i ruoli nel passaggio 3.

  Per trovare il numero ID dell' AWS account utilizzando AWS Management Console, scegli Supporto nella barra di navigazione in alto a destra, quindi scegli Support Center. Il numero dell'account (ID) viene visualizzato nel riquadro di navigazione a sinistra.



- Creazione e modifica di utenti, ruoli e policy IAM nella AWS Management Console. Tuttavia, se hai bisogno di aiuto per ricordare un processo di gestione IAM, questo tutorial fornisce link dove puoi visualizzare step-by-step le istruzioni.

Fase 1: creazione degli utenti di test

A scopo di test, crea quattro utenti IAM con le autorizzazioni per assumere ruoli con gli stessi tag. In questo modo è più facile aggiungere più utenti ai team. Quando si associano i tag agli utenti, questi ottengono automaticamente l'accesso per assumere il ruolo corretto. Non è necessario aggiungere gli utenti alla policy di trust del ruolo se lavorano su un solo progetto e in un solo team.

1. Creare la seguente policy gestita dal cliente denominata `access-assume-role`. Per ulteriori informazioni sulla creazione della policy JSON, consulta Creazione di policy IAM.

   Policy ABAC: assumere qualsiasi ruolo ABAC, ma solo quando i tag utente e ruolo corrispondono

La policy seguente consente a un utente di assumere qualsiasi ruolo nell'account con il prefisso `access-` nel nome. Il ruolo deve inoltre essere taggato con gli stessi tag di progetto, team e centro di costi dell'utente.

Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialAssumeRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::account-ID-without-hyphens:role/access-*",
            "Condition": {
                "StringEquals": {
                    "iam:ResourceTag/access-project": "${aws:PrincipalTag/access-
project}",
                    "iam:ResourceTag/access-team": "${aws:PrincipalTag/access-
team}",
                    "iam:ResourceTag/cost-center": "${aws:PrincipalTag/cost-
center}"
                }
            }
        }
    ]
}
```

Per ridimensionare questa esercitazione a un numero elevato di utenti, è possibile collegare la policy a un gruppo e aggiungere ogni utente al gruppo. Per ulteriori informazioni, consulta Creazione di gruppi di utenti IAM e Aggiunta e rimozione di utenti in un gruppo di utenti IAM.

2. Creare i seguenti utenti IAM e collegare la policy di autorizzazione `access-assume-role`. Assicurarsi di selezionare Fornire l'accesso agli utenti a AWS Management Console, poi aggiungere i seguenti tag. Per ulteriori informazioni sulla creazione e l'assegnazione di tag di un nuovo utente, consulta Creazione di utenti IAM (console).

Utenti ABAC

Nome utente	Chiave tag utente	Valore tag utente
access-Arnav-peg-eng	access-project access-team cost-center	peg eng 987654
access-Mary-peg-qas	access-project access-team cost-center	peg qas 987654
access-Saanvi-uni-eng	access-project access-team cost-center	uni eng 123456
access-Carlos-uni-qas	access-project access-team cost-center	uni qas 123456

Fase 2: creazione della policy ABAC

Creare la seguente policy denominata **access-same-project-team**. Questa policy verrà aggiunta ai ruoli in un passaggio successivo. Per ulteriori informazioni sulla creazione della policy JSON, consulta Creazione di policy IAM.

Per ulteriori policy che è possibile adattare a questa esercitazione, vedere le pagine seguenti:

- Controllo dell'accesso per i principali IAM
- Amazon EC2: consente l'avvio o l'arresto di istanze EC2 che un utente ha contrassegnato, a livello programmatico e nella console
- EC2: avvio o arresto di istanze in base alla corrispondenza dei tag della risorsa e del principale

- EC2: avvio o arresto di istanze in base ai tag
- IAM: assumere ruoli che dispongono di un tag specifico

Policy ABAC: accesso alle risorse di Secrets Manager solo quando il tag del principale e quello della risorsa corrispondono

La policy seguente consente ai principali di creare, leggere, modificare ed eliminare risorse, ma solo quando tali risorse sono contrassegnate con le stesse coppie chiave-valore del principale. Quando un principale crea una risorsa, deve aggiungere i tag `access-project`, `access-team` e `cost-center` con valori corrispondenti ai tag del principale. La policy consente anche l'aggiunta di tag facoltativi `Name` o `OwnedBy`.

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
         "Sid": "AllActionsSecretsManagerSameProjectSameTeam",
         "Effect": "Allow",
         "Action": "secretsmanager:*",
         "Resource": "*",
         "Condition": {
             "StringEquals": {
                 "aws:ResourceTag/access-project": "${aws:PrincipalTag/access-
project}",
                 "aws:ResourceTag/access-team": "${aws:PrincipalTag/access-team}",
                 "aws:ResourceTag/cost-center": "${aws:PrincipalTag/cost-center}"
             },
             "ForAllValues:StringEquals": {
                 "aws:TagKeys": [
                     "access-project",
                     "access-team",
                     "cost-center",
                     "Name",
                     "OwnedBy"
                 ]
             },
             "StringEqualsIfExists": {
                 "aws:RequestTag/access-project": "${aws:PrincipalTag/access-project}",
                 "aws:RequestTag/access-team": "${aws:PrincipalTag/access-team}",
                 "aws:RequestTag/cost-center": "${aws:PrincipalTag/cost-center}"
             }
```

```
        }
    },
    {
        "Sid": "AllResourcesSecretsManagerNoTags",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetRandomPassword",
            "secretsmanager:ListSecrets"
        ],
        "Resource": "*"
    },
    {
        "Sid": "ReadSecretsManagerSameTeam",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:Describe*",
            "secretsmanager:Get*",
            "secretsmanager:List*"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/access-team": "${aws:PrincipalTag/access-team}"
            }
        }
    },
    {
        "Sid": "DenyUntagSecretsManagerReservedTags",
        "Effect": "Deny",
        "Action": "secretsmanager:UntagResource",
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringLike": {
                "aws:TagKeys": "access-*"
            }
        }
    },
    {
        "Sid": "DenyPermissionsManagement",
        "Effect": "Deny",
        "Action": "secretsmanager:*Policy",
        "Resource": "*"
    }
]
```

```
}
```

Che cosa fa questa policy?

- L'istruzione `AllActionsSecretsManagerSameProjectSameTeam` consente tutte le operazioni relative a questo servizio su tutte le risorse correlate, ma solo se i tag di risorsa corrispondono ai tag del principale. Aggiungendo `"Action": "secretsmanager:*"` alla policy, la policy stessa cresce man mano che Secrets Manager cresce. Se Secrets Manager aggiunge una nuova operazione API, non è necessario aggiungere tale operazione all'istruzione. L'istruzione implementa ABAC utilizzando tre blocchi di condizione. La richiesta è consentita solo se tutti e tre i blocchi restituiscono true.
  - Il primo blocco di condizione di questa istruzione restituisce true se le chiavi tag specificate sono presenti nella risorsa e i loro valori corrispondono ai tag del principale. Questo blocco restituisce false per i tag non corrispondenti o per le operazioni che non supportano il tag delle risorse. Per sapere quali azioni non sono consentite da questo blocco, consulta Azioni, risorse e chiavi di condizione per AWS Secrets Manager. Questa pagina mostra che le operazioni eseguite sul tipo di risorsa Segreto supportano la chiave di condizione `secretsmanager:ResourceTag/tag-key`. Alcune azioni di Secrets Manager non supportano tale tipo di risorsa, inclusi `GetRandomPassword` e `ListSecrets`. Per consentire tali azioni, è necessario creare istruzioni aggiuntive.
  - Il secondo blocco di condizione restituisce true se ogni chiave del tag passata nella richiesta è incluso nell'elenco specificato. Questo viene fatto utilizzando `ForAllValues` con l'operatore condizionale `StringEquals`. Se non vengono passate chiavi o un sottoinsieme del set di chiavi, la condizione restituisce true. Ciò consente operazioni `Get*` che non consentono il passaggio di tag nella richiesta. Se il richiedente include una chiave del tag che non è nell'elenco, la condizione restituisce false. Ogni chiave dei tag passata nella richiesta deve corrispondere a un elemento di tale elenco. Per ulteriori informazioni, consulta Chiavi di contesto multivalore.
  - Il terzo blocco di condizioni restituisce true se la richiesta supporta il passaggio dei tag, se tutti e tre i tag sono presenti e se corrispondono ai valori del tag del principale. Questo blocco restituisce true anche se la richiesta non supporta il passaggio di tag. Tale risultato è ottenuto tramite l'utilizzo di `...IfExists` nell'operatore condizionale. Il blocco restituisce false se non vi è alcun tag passato durante un'operazione che li supporta o se le chiavi e i valori dei tag non corrispondono.
- L'istruzione `AllResourcesSecretsManagerNoTags` permette le operazioni `GetRandomPassword` e `ListSecrets` che non sono consentite dalla prima istruzione.

- L'istruzione `ReadSecretsManagerSameTeam` permette le operazioni di sola lettura se il principale è contrassegnato con lo stesso tag di team di accesso della risorsa. Ciò è consentito indipendentemente dal progetto o dal tag del centro di costi.
- L'istruzione `DenyUntagSecretsManagerReservedTags` rifiuta le richieste di rimuovere da Secrets Manager i tag con chiavi che iniziano con il prefisso "access-". Questi tag vengono utilizzati per controllare l'accesso alle risorse, pertanto la rimozione dei tag potrebbe rimuovere le autorizzazioni.
- L'istruzione `DenyPermissionsManagement` nega l'accesso per creare, modificare o eliminare policy basate sulle risorse di Secrets Manager. Queste policy possono essere utilizzate per modificare le autorizzazioni dei segreti.

> **Important**
>
> Questa policy utilizza una strategia per consentire tutte le operazioni per un servizio, ma negando esplicitamente le operazioni di modifica delle autorizzazioni. Il rifiuto di un'operazione sostituisce qualsiasi altra policy che consente al principale di eseguire tale operazione. Ciò può avere risultati imprevisti. Come best practice, usare il rifiuto esplicito solo quando non vi è alcuna circostanza in cui debba essere consentita tale operazione. In caso contrario, consentire un elenco di singole operazioni in modo che le operazioni indesiderate vengano negate per impostazione predefinita.

Fase 3: creazione di ruoli

Crea i seguenti ruoli IAM e collega la policy **`access-same-project-team`** creata nella fase precedente. Per ulteriori informazioni sulla creazione dei ruoli IAM, consulta Creazione di un ruolo per delegare le autorizzazioni a un utente IAM. Se decidi di utilizzare la federazione anziché gli utenti e i ruoli IAM, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

Ruoli ABAC

Funzione processo	Nome ruolo	Tag di ruolo	Descrizione del ruolo
Progetto Pegasus Engineering	`access-peg-engineering`	`access-project = peg`	Consente agli ingegneri di leggere tutte le risorse ingegneristiche e di creare

Funzione processo	Nome ruolo	Tag di ruolo	Descrizione del ruolo
		`access-team = eng` `cost-center = 987654`	e gestire le risorse ingegneristiche di Pegasus.

Progetto Pegasus Quality Assurance	`access-peg-quality-assurance`	`access-project = peg`
		`access-team = qas`
		`cost-center = 987654`
			Consente al team QA di leggere tutte le risorse di QA e di creare e gestire tutte le risorse QA di Pegasus.

Progetto Unicorn Engineering	`access-uni-engineering`	`access-project = uni`
		`access-team = eng`
		`cost-center = 123456`
			Consente agli ingegneri di leggere tutte le risorse ingegneristiche e creare e gestire le risorse ingegneristiche di Unicorn.

Funzione processo	Nome ruolo	Tag di ruolo	Descrizione del ruolo

Progetto Unicorn Quality Assurance	`access-uni-quality-assurance`	`access-project = uni`
		
		`access-team = qas`
		
		`cost-center = 123456`
			Consente al team QA di leggere tutte le risorse QA e di creare e gestire tutte le risorse QA di Unicorn.

Fase 4: verifica della creazione di segreti

La policy di autorizzazione collegata ai ruoli consente ai dipendenti di creare segreti. Questo è consentito solo se il segreto è contrassegnato con il relativo progetto, team e centro di costi. Verifica che le autorizzazioni funzionino come previsto accedendo come i tuoi utenti, assumendo il ruolo corretto e testando l'attività in Secrets Manager.

Per testare la creazione di un segreto con e senza i tag richiesti

1. Nella finestra principale del browser, resta connesso come utente amministratore in modo da poter esaminare utenti, ruoli e policy in IAM. Utilizzare una finestra di navigazione in incognito del browser o un browser separato per i test. Lì, accedi come utente IAM `access-Arnav-peg-eng` e apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Provare a passare al ruolo `access-uni-engineering`.

   Questa operazione ha esito negativo perché i valori de tag `access-project` e `cost-center` non corrispondono all'utente `access-Arnav-peg-eng` e al ruolo `access-uni-engineering`.

   Per ulteriori informazioni sul cambio di ruolo in AWS Management Console, vedere Cambio di un ruolo (console)
3. Passare al ruolo `access-peg-engineering`.

4. Archiviare un nuovo segreto utilizzando le seguenti informazioni. Per informazioni su come archiviare un segreto, consulta Creazione di un segreto di base nella Guida per l'utente di AWS Secrets Manager .

> **Important**
>
> Secrets Manager visualizza avvisi che segnalano che non disponi delle autorizzazioni per i servizi AWS aggiuntivi che funzionano con Secrets Manager. Ad esempio, per creare credenziali per un database Amazon RDS, è necessario disporre dell'autorizzazione per descrivere istanze RDS, cluster RDS e cluster Amazon Redshift. Puoi ignorare questi avvisi poiché non stai utilizzando questi AWS servizi specifici in questo tutorial.

1. Nella sezione Seleziona tipo di segreto, scegliere Altro tipo di segreti. Nelle due caselle di testo, inserire `test-access-key` e `test-access-secret`.
2. Nel campo Nome segreto inserire il valore `test-access-peg-eng`.
3. Aggiungere diverse combinazioni di tag dalla tabella seguente e visualizzare il comportamento previsto.
4. Scegliere Memorizza per provare a creare il segreto. Se l'archiviazione non riesce, torna alle pagine della console di Secrets Manager precedenti e utilizza il set di tag successivo dalla tabella seguente. L'ultimo set di tag è consentito e creerà con successo il segreto.

Combinazioni di tag ABAC per il ruolo **`test-access-peg-eng`**

Valore tag **`access-project`**	Valore tag **`access-team`**	Valore tag **`cost-center`**	Tag aggiuntivi	Comportamento previsto
(nessuno)	(nessuno)	(nessuno)	(nessuno)	Negato perché il valore del tag `access-project` non corrisponde al valore del ruolo di `peg`.
`uni`	`eng`	`987654`	(nessuno)	Negato perché il valore del tag `access-project` non corrisponde al valore del ruolo di `peg`.

Valore tag **access-project**	Valore tag **access-team**	Valore tag **cost-center**	Tag aggiuntivi	Comportamento previsto
peg	qas	987654	(nessuno)	Negato perché il valore del tag access-team non corrisponde al valore del ruolo di eng.
peg	eng	123456	(nessuno)	Negato perché il valore del tag cost-center non corrisponde al valore del ruolo di 987654.
peg	eng	987654	owner = Jane	Negato perché il tag aggiuntiv o owner non è consentito dalla policy, anche se tutti e tre i tag richiesti sono presenti e i relativi valori corrispondono ai valori del ruolo.
peg	eng	987654	Name = Jane	Consentito perché tutti e tre i tag richiesti sono presenti e i loro valori corrispondono ai valori del ruolo. È anche possibile includere il tag opzionale Name.

5. Disconnettersi e ripetere i primi tre passaggi di questa procedura per ciascuno dei seguenti ruoli e valori dei tag. Nel quarto passaggio di questa procedura, verificare tutti i set di tag mancanti, tag facoltativi, tag non consentiti e valori di tag non validi selezionati. Quindi utilizzare i tag richiesti per creare un segreto con i seguenti tag e nome.

   Ruoli e tag ABAC

Nome utente	Nome ruolo	Nome segreto	Tag segreto
access-Mary-peg-qas	access-peg-quality-assurance	test-access-peg-qas	access-project = peg access-team = qas

Nome utente	Nome ruolo	Nome segreto	Tag segreto
			`cost-center = 987654`

`access-Saanvi-uni-eng`	`access-uni-engineering`	`test-access-uni-eng`	`access-project = uni`
			`access-team = eng`
			`cost-center = 123456`

`access-Carlos-uni-qas`	`access-uni-quality-assurance`	`test-access-uni-qas`	`access-project = uni`
			`access-team = qas`
			`cost-center = 123456`

Fase 5: verifica della visualizzazione dei segreti

La policy collegata a ciascun ruolo consente ai dipendenti di visualizzare eventuali segreti contrassegnati con il nome del team, indipendentemente dal progetto. Verifica che le autorizzazioni funzionino come previsto testando i ruoli in Secrets Manager.

Per testare la visualizzazione di un segreto con e senza i tag richiesti

1. Accedi come uno dei seguenti utenti IAM:
   - `access-Arnav-peg-eng`
   - `access-Mary-peg-qas`
   - `access-Saanvi-uni-eng`
   - `access-Carlos-uni-qas`
2. Passa al ruolo corrispondente:

- `access-peg-engineering`
- `access-peg-quality-assurance`
- `access-uni-engineering`
- `access-uni-quality-assurance`

Per ulteriori informazioni sul cambio di ruolo in AWS Management Console, vedereCambio di un ruolo (console).

3. Nel riquadro di navigazione a sinistra, scegli l'icona del menu per espanderlo, quindi scegliere Segreti.
4. Dovrebbero essere visualizzati tutti e quattro i segreti nella tabella, indipendentemente dal proprio ruolo attuale. Ciò accade perché la policy denominata `access-same-project-team` consente l'operazione `secretsmanager:ListSecrets` per tutte le risorse.
5. Scegli il nome di uno dei segreti.
6. Nella pagina dei dettagli del segreto, i tag del ruolo determinano se è possibile visualizzare il contenuto della pagina. Confrontare il nome del proprio ruolo con il nome del segreto. Se condividono lo stesso nome del team, i tag `access-team` corrispondono. Se non corrispondono, l'accesso viene negato.

Comportamento di visualizzazione del segreto ABAC per ogni ruolo

Nome ruolo	Nome segreto	Comportamento previsto
`access-peg-engineering`	`test-access-peg-eng`	Consentito
	`test-access-peg-qas`	Negato
	`test-access-uni-eng`	Consentito
	`test-access-uni-qas`	Negato
`access-peg-quality-assurance`	`test-access-peg-eng`	Negato
	`test-access-peg-qas`	Consentito
	`test-access-uni-eng`	Negato
	`test-access-uni-qas`	Consentito

Nome ruolo	Nome segreto	Comportamento previsto
`access-uni-engineering`	`test-access-peg-eng`	Consentito
	`test-access-peg-qas`	Negato
	`test-access-uni-eng`	Consentito
	`test-access-uni-qas`	Negato
`access-uni-quality-assurance`	`test-access-peg-eng`	Negato
	`test-access-peg-qas`	Consentito
	`test-access-uni-eng`	Negato
	`test-access-uni-qas`	Consentito

7. Nel percorso di navigazione nella parte superiore della pagina, scegliere Segreti per tornare all'elenco dei segreti. Ripetere i passaggi di questa procedura utilizzando ruoli diversi per verificare se è possibile visualizzare ciascuno dei segreti.

Fase 6: verifica della scalabilità

Un motivo importante per utilizzare il controllo degli accessi basato su attributi (ABAC) invece del controllo di accesso basato su ruoli (RBAC) è la scalabilità. Man mano che l'azienda aggiunge nuovi progetti, team o persone AWS, non è necessario aggiornare le politiche basate su ABAC. Ad esempio, si supponga che Example Company stia finanziando un nuovo progetto dal nome in codice Centaur. Un ingegnere di nome Saanvi Sarkar sarà l'ingegnere responsabile di Centaur continuando a lavorare al progetto Unicorn . Saanvi esaminerà anche i lavori per il progetto Peg. Ci sono anche diversi ingegneri appena assunti, tra cui Nikhil Jayashankar, che lavoreranno solo al progetto Centaur .

Per aggiungere il nuovo progetto a AWS

1. Accedi come utente amministratore IAM e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione sulla sinistra, scegli Ruoli e aggiungi un ruolo IAM denominato `access-cen-engineering`. Collega la policy delle autorizzazioni **`access-same-project-team`** al ruolo e aggiungere i seguenti tag di ruolo:

   - `access-project` = `cen`
   - `access-team` = `eng`
   - `cost-center` = `101010`

3. Nel riquadro di navigazione sinistro, scegli Utenti.
4. Aggiungi un nuovo utente denominato `access-Nikhil-cen-eng`, collega la policy denominata `access-assume-role` e aggiungi i seguenti tag utente.

   - `access-project` = `cen`
   - `access-team` = `eng`
   - `cost-center` = `101010`

5. Utilizza le procedure in Fase 4: verifica della creazione di segreti e Fase 5: verifica della visualizzazione dei segreti. In un'altra finestra del browser, verifica che Nikhil possa creare segreti solo per il team di ingegneria di Centaur e che possa visualizzare tutti i segreti dei team di ingegneria.
6. Nella finestra principale del browser in cui hai effettuato l'accesso come amministratore, scegli l'utente `access-Saanvi-uni-eng`.
7. Nella scheda Autorizzazioni, rimuovi la politica delle access-assume-roleautorizzazioni.
8. Aggiungi la seguente policy inline denominata `access-assume-specific-roles`. Per ulteriori informazioni sull'aggiunta di una policy inline a un utente, consulta Per incorporare una policy inline per un utente o un ruolo (console).

   Policy ABAC: assunzione dei soli ruoli specifici

   Questa policy consente a Saanvi di assumere i ruoli ingegneristici per i progetti Pegasus e Centaur. È necessario creare questa policy personalizzata perché IAM non supporta tag multivalore. Non è possibile etichettare l'utente di Saanvi con `access-project` = `peg` e `access-project` = `cen`. Inoltre, il modello di AWS autorizzazione non può corrispondere a entrambi i valori. Per ulteriori informazioni, consulta Regole per l'etichettatura in IAM e AWS STS. È invece necessario specificare manualmente i due ruoli che può assumere.

   Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialAssumeSpecificRoles",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::account-ID-without-hyphens:role/access-peg-
engineering",
                "arn:aws:iam::account-ID-without-hyphens:role/access-cen-
engineering"
            ]
        }
    ]
}
```

9. Utilizza le procedure in Fase 4: verifica della creazione di segreti e Fase 5: verifica della visualizzazione dei segreti. In un'altra finestra del browser, conferma che Saanvi possa assumere entrambi i ruoli. Verifica che sia in grado di creare segreti solo per i suoi progetto, team e centro di costo, a seconda dei tag del ruolo. Verifica anche che possa visualizzare i dettagli su eventuali segreti di proprietà del team di ingegneria, inclusi quelli che ha appena creato.

Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti

La policy `access-same-project-team` collegata ai ruoli consente ai dipendenti di aggiornare ed eliminare eventuali segreti contrassegnati con il proprio progetto, team e centro costi. Verifica che le autorizzazioni funzionino come previsto testando i ruoli in Secrets Manager.

Per verificare l'aggiornamento e l'eliminazione di un segreto con e senza i tag richiesti

1. Accedi come uno dei seguenti utenti IAM:

   - `access-Arnav-peg-eng`
   - `access-Mary-peg-qas`
   - `access-Saanvi-uni-eng`
   - `access-Carlos-uni-qas`

- `access-Nikhil-cen-eng`

2. Passa al ruolo corrispondente:

   - `access-peg-engineering`
   - `access-peg-quality-assurance`
   - `access-uni-engineering`
   - `access-peg-quality-assurance`
   - `access-cen-engineering`

   Per ulteriori informazioni sul cambio di ruolo in AWS Management Console, vedereCambio di un ruolo (console).

3. Per ogni ruolo, prova ad aggiornare la descrizione del segreto e quindi prova a eliminare i seguenti segreti. Per ulteriori informazioni, consulta Modifica di un segreto ed Eliminazione e ripristino di un segreto nella Guida per l'utente di AWS Secrets Manager .

   Comportamento di aggiornamento ed eliminazione dei segreti ABAC per ogni ruolo

Nome ruolo	Nome segreto	Comportamento previsto
`access-peg-engineering`	`test-access-peg-eng`	Consentito
	`test-access-uni-eng`	Negato
	`test-access-uni-qas`	Negato
`access-peg-quality-assurance`	`test-access-peg-qas`	Consentito
	`test-access-uni-eng`	Negato
`access-uni-engineering`	`test-access-uni-eng`	Consentito
	`test-access-uni-qas`	Negato
`access-peg-quality-assurance`	`test-access-uni-qas`	Consentito

Riepilogo

Tutte le fasi necessarie per utilizzare i tag per il controllo degli accessi basato su attributi (ABAC) sono state completate correttamente. L'utente ha imparato a definire una strategia di tagging. Tale strategia è stata applicata alle proprie entità e risorse. È stata creata e applicata una policy che impone la strategia a Secrets Manager. L'utente ha anche imparato che ABAC è facilmente ridimensionabile nel momento in cui si aggiungono nuovi progetti e membri del team. Di conseguenza, sarai in grado di accedere alla console IAM con i ruoli di test e scoprire come utilizzare i tag per ABAC in AWS.

> **Note**
>
> L'utente ha aggiunto policy che consentono operazioni solo in condizioni specifiche. Se si applica un criterio diverso agli utenti o ai ruoli con autorizzazioni più ampie, è possibile che le azioni non siano limitate a richiedere l'assegnazione di tag. Ad esempio, se concedi a un utente autorizzazioni amministrative complete utilizzando la politica `AdministratorAccess` AWS gestita, tali politiche non limiteranno tale accesso. Per ulteriori informazioni su come vengono determinate le autorizzazioni quando sono coinvolte più policy, vedere Determinazione se una richiesta è consentita o rifiutata in un account.

Risorse correlate

Per informazioni correlate, consulta le seguenti risorse:

- A cosa serve ABAC? AWS
- AWS chiavi di contesto della condizione globale
- Creazione di utenti IAM (console)
- Creazione di un ruolo per delegare le autorizzazioni a un utente IAM
- Tagging delle risorse IAM
- Controllo dell'accesso alle AWS risorse tramite tag
- Cambio di un ruolo (console)
- Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC

Per sapere come monitorare i tag nel tuo account, consulta Monitorare le modifiche ai tag sulle AWS risorse con flussi di lavoro serverless e Amazon CloudWatch Events.

Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag alle risorse IAM, incluse le entità IAM (utenti o ruoli) e alle AWS risorse. Quando le entità vengono utilizzate per effettuare richieste a AWS, diventano principali e tali entità includono i tag.

È inoltre possibile passare i tag di sessione quando si assume un ruolo o si federa un utente. È quindi possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni alle entità sulla base dei relativi tag. Quando si utilizzano i tag per controllare l'accesso alle risorse AWS , si consente ai team e alle risorse di crescere con la necessità di un minor numero di modifiche alle policy AWS . Le politiche ABAC sono più flessibili delle AWS politiche tradizionali, che richiedono di elencare ogni singola risorsa. Per ulteriori informazioni su ABAC e i suoi vantaggi rispetto alle policy tradizionali, consulta A cosa serve ABAC? AWS.

Se, per gestire le identità degli utenti aziendali, l'azienda utilizza un provider di identità basato su SAML (IdP) è possibile utilizzare gli attributi SAML per il controllo degli accessi AWS a granularità fine. Gli attributi possono includere identificatori del centro di costo, indirizzi e-mail degli utenti, reparti di appartenenza e assegnazioni di progetto. Quando si passano questi attributi come tag di sessione, è quindi possibile controllare l'accesso ad AWS in base a tali tag di sessione.

Per completare l'esercitazione su ABAC passando gli attributi SAML all'entità sessione, completare le attività descritte in Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag, con le modifiche incluse in questa sezione.

Prerequisiti

Per eseguire la procedura che prevede l'utilizzo dei tag di sessione SAML per ABAC, è necessario disporre preventivamente quanto segue:

- Accesso a un IdP basato su SAML in cui è possibile creare utenti di test con attributi specifici.
- La possibilità di effettuare l'accesso come utente con autorizzazioni di amministratore.
- Creazione e modifica di utenti, ruoli e policy IAM nella AWS Management Console. Tuttavia, se hai bisogno di aiuto per ricordare un processo di gestione IAM, il tutorial ABAC fornisce collegamenti in cui puoi visualizzare le istruzioni. step-by-step
- Completa la configurazione di un IdP basato su SAML in IAM. Per visualizzare maggiori dettagli e i collegamenti alla documentazione IAM dettagliata, consulta Passaggio dei tag di sessione tramite SAML AssumeRoleWith.

Fase 1: creazione degli utenti di test

Seguire le istruzioni in Fase 1: creazione degli utenti di test. Poiché le identità sono definite dal provider, non è necessario aggiungere gli utenti IAM per i propri dipendenti.

Fase 2: creazione della policy ABAC

Per creare la policy gestita specificata in IAM, seguire le istruzioni riportate in Fase 2: creazione della policy ABAC.

Fase 3: creazione e configurazione del ruolo SAML

Quando utilizzi il tutorial ABAC per SAML, devi eseguire passaggi aggiuntivi per creare il ruolo, configurare l'IdP SAML e abilitare l'accesso. AWS Management Console Per ulteriori informazioni, consulta Fase 3: creazione di ruoli.

Fase 3A: creazione del ruolo SAML

Creare un singolo ruolo in relazione di trust con il provider di identità SAML e l'utente `test-session-tags` creato nel passaggio 1. L'esercitazione ABAC utilizza ruoli separati con diversi tag di ruolo. Poiché si stanno passando i tag di sessione dal proprio IdP SAML, c'è bisogno di un solo ruolo. Per informazioni su come creare un ruolo basato su SAML, consulta Creare un ruolo per la federazione SAML 2.0 (console).

Denomina il ruolo `access-session-tags`. Collegare la policy di autorizzazione `access-same-project-team` al ruolo. Modificare la policy di trust al fine di utilizzare la policy riportata di seguito. Per istruzioni dettagliate su come modificare la relazione di trust di un ruolo, consulta Modifica di un ruolo (console).

La seguente policy di trust del ruolo consente al provider di identità SAML e all'utente `test-session-tags` di assumere il ruolo. Al momento dell'assunzione del ruolo, è necessario passare i tre tag di sessione specificati. L'operazione `sts:TagSession` è necessaria per consentire il passaggio dei tag di sessione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSamlIdentityAssumeRole",
            "Effect": "Allow",
            "Action": [
```

```
                "sts:AssumeRoleWithSAML",
                "sts:TagSession"
            ],
            "Principal": {"Federated":"arn:aws:iam::123456789012:saml-
provider/ExampleCorpProvider"},
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/cost-center": "*",
                    "aws:RequestTag/access-project": "*",
                    "aws:RequestTag/access-team": [
                        "eng",
                        "qas"
                    ]
                },
                "StringEquals": {"SAML:aud": "https://signin.aws.amazon.com/saml"}
            }
        }
    ]
}
```

La `AllowSamlIdentityAssumeRole` dichiarazione consente ai membri dei team di ingegneria e controllo qualità di assumere questo ruolo quando si uniscono all'Example Corporation AWS IdP. Il provider SAML `ExampleCorpProvider` è definito in IAM. L'amministratore ha già impostato l'asserzione SAML per passare i tre tag di sessione richiesti. L'asserzione può passare tag aggiuntivi, ma questi tre devono essere presenti. Gli attributi dell'identità possono presentare qualsiasi valore per i tag `access-project` e `cost-center`. Tuttavia, il valore dell'attributo `access-team` deve corrispondere a `eng` o `qas` a indicare che l'identità corrisponde al team di Engineering o di Quality Assurance.

Passaggio 3B: configurazione dell'IdP SAML

Configurare l'IdP SAML affinché passi gli attributi `cost-center`, `access-project` e `access-team` come tag di sessione. Per ulteriori informazioni, consulta Passaggio dei tag di sessione tramite SAML AssumeRoleWith.

Per passare questi attributi come tag di sessione, includere i seguenti elementi nell'asserzione SAML.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:cost-center">
  <AttributeValue>987654</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:access-project">
  <AttributeValue>peg</AttributeValue>
```

```
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:access-team">
  <AttributeValue>eng</AttributeValue>
</Attribute>
```

Fase 3C: attivazione dell'accesso alla console

Abilita l'accesso alla console per gli utenti SAML federati. Per ulteriori informazioni, consulta Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console.

Fase 4: verifica della creazione di segreti

Unisciti all'utilizzo del ruolo. AWS Management Console `access-session-tags` Per ulteriori informazioni, consulta Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console. Quindi seguire le istruzioni in Fase 4: verifica della creazione di segreti per creare segreti. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC. Per ulteriori informazioni, consulta Fase 4: verifica della creazione di segreti.

Fase 5: verifica della visualizzazione dei segreti

Seguire le istruzioni descritte in Fase 5: verifica della visualizzazione dei segreti per visualizzare i segreti creati nel passaggio precedente. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC.

Fase 6: verifica della scalabilità

Seguire le istruzioni descritte in Fase 6: verifica della scalabilità per testare la scalabilità. Eseguire questa operazione aggiungendo una nuova identità nel proprio IdP basato su SAML con i seguenti attributi:

- `cost-center = 101010`
- `access-project = cen`
- `access-team = eng`

Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti

Seguire le istruzioni descritte in Fase 7: verifica dell'aggiornamento e dell'eliminazione dei segreti per aggiornare ed eliminare i segreti. Utilizzare diverse identità SAML con attributi da abbinare ai tag indicati nell'esercitazione ABAC.

> **Important**
>
> Eliminare tutti i segreti creati per evitare addebiti in fattura. Per informazioni sui prezzi di Secrets Manager, consulta Prezzi di AWS Secrets Manager.

Riepilogo

Sono stati completati tutti i passaggi necessari per utilizzare i tag di sessione SAML e i tag delle risorse per la gestione delle autorizzazioni.

> **Note**
>
> L'utente ha aggiunto policy che consentono operazioni solo in condizioni specifiche. Se si applica un criterio diverso agli utenti o ai ruoli con autorizzazioni più ampie, è possibile che le azioni non siano limitate a richiedere l'assegnazione di tag. Ad esempio, se concedi a un utente autorizzazioni amministrative complete utilizzando la policy `AdministratorAccess` AWS gestita, queste politiche non limiteranno tale accesso. Per ulteriori informazioni su come vengono determinate le autorizzazioni quando sono coinvolte più policy, vedere Determinazione se una richiesta è consentita o rifiutata in un account.

Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA

Puoi consentire agli utenti di gestire i propri dispositivi e credenziali di autenticazione a più fattori (MFA) nella pagina Credenziali di sicurezza. È possibile utilizzare la AWS Management Console per configurare le credenziali (chiavi di accesso, password, certificati di firma e chiavi pubbliche SSH), eliminare o disattivare le credenziali superflue e abilitare i dispositivi MFA per gli utenti. Sebbene sia utile per un numero ridotto di utenti, è un'operazione che potrebbe richiedere molto tempo se il numero di utenti cresce. Mostrare come abilitare queste best practice senza sovraccaricare gli amministratori è l'obiettivo di questo tutorial.

Questo tutorial mostra come consentire agli utenti di accedere ai AWS servizi, ma solo quando accedono con MFA. Se non sono registrati con un dispositivo MFA, gli utenti non possono accedere ad altri servizi.

Questo flusso di lavoro ha tre fasi di base.

[Fase 1: creazione di una policy per applicare l'accesso MFA]

Crea una policy gestita dal cliente che impedisce tutte le azioni eccetto le poche operazioni IAM. Queste eccezioni consentono a un utente di modificare le proprie credenziali e gestire i propri dispositivi MFA nella pagina Credenziali di sicurezza. Per ulteriori informazioni sull'accesso alla pagina, consulta Come gli utenti IAM possono cambiare le proprie password (console).

[Fase 2: Collegamento delle policy al gruppo di utenti di test]

Crea un gruppo di utenti i cui membri hanno accesso completo a tutte le operazioni Amazon EC2 se effettuano l'accesso con MFA. Per creare un gruppo di utenti di questo tipo, è necessario allegare sia la politica AWS gestita richiamata `AmazonEC2FullAccess` sia la politica gestita dal cliente creata nel primo passaggio.

[Fase 3: test dell'accesso dell'utente]

Accedi come utente di prova per verificare che l'accesso ad Amazon EC2 sia bloccato fino a quando l'utente non crea un dispositivo MFA. L'utente può quindi accedere utilizzando tale dispositivo.

Prerequisiti

Per eseguire queste fasi in questo tutorial, è necessario quanto segue:

- E a Account AWS cui puoi accedere come utente IAM con autorizzazioni amministrative.
- Il numero ID dell'account, che si digita nella policy nella Fase 1.

  Per trovare il numero ID dell'account nella barra di navigazione in alto sulla pagina, selezionare Support (Supporto) e selezionare Support Center (Centro di supporto). Puoi trovare l'ID dell'account nel menu Supporto di questa pagina.
- Un dispositivo MFA virtuale (basato su software), una chiave di sicurezza FIDO o un dispositivo MFA basato su hardware.
- Un utente IAM di prova che è membro di un gruppo come segue:

Create user (Crea utente)		Creazione e configurazione di account di gruppo di utenti		
Nome utente	Altre istruzioni	Nome gruppo di utenti	Aggiungere utente come un membro	Altre istruzioni
MFAUser	Seleziona solo l'opzione per Enable console access – optional (Abilita , l'accesso alla console - facoltativo) e assegna una password.	EC2MFA	MFAUser	NON collegare policy o concedere autorizzazioni a questo gruppo di utenti.

Fase 1: creazione di una policy per applicare l'accesso MFA

Per iniziare, crea una policy gestita dal cliente IAM che nega tutte le autorizzazioni tranne quelle richieste per gli utenti IAM per gestire le credenziali e i dispositivi MFA.

1. Accedi alla console di AWS gestione come utente con credenziali di amministratore. Per aderire alle best practice di IAM, non accedere con le tue Utente root dell'account AWS credenziali.

   > **Important**
   >
   > Le migliori pratiche IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché utilizzare utenti IAM con credenziali a lungo termine.

2. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

3. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.

4. Selezionare la scheda JSON e copiare il testo dal documento della seguente policy JSON: AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

5. Incolla il testo della policy nella casella di testo JSON. Risolvi eventuali avvisi di sicurezza, errori o avvertenze generali generati durante la convalida delle policy, quindi scegli Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Tuttavia, la policy qui sopra include l'elemento `NotAction`, che non è supportato nell'editor visivo. Per questa policy, verrà visualizzata una notifica nella scheda Visual Editor (Editor visivo). Torna alla scheda JSON per continuare a lavorare con questa policy.
   > Questo esempio di policy non consente agli utenti di reimpostare la password durante il primo accesso a AWS Management Console . Ti consigliamo di non concedere autorizzazioni ai nuovi utenti fino a quando non hanno effettuato l'accesso e reimpostato la password.

6. Nella pagina Verifica policy, digita **`Force_MFA`** come nome della policy. Per la descrizione della policy, digita **`This policy allows users to manage their own passwords and MFA devices but nothing else unless they authenticate with MFA.`**. Nell'area Tag, puoi facoltativamente aggiungere coppie chiave-valore di tag alla policy gestita dal cliente. Esamina le autorizzazioni concesse dalla policy, quindi scegli Crea policy per salvare il lavoro.

   La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

Fase 2: Collegamento delle policy al gruppo di utenti di test

In seguito collega le due policy al gruppo di utenti IAM di test, che verrà utilizzato per concedere le autorizzazioni protette mediante MFA.

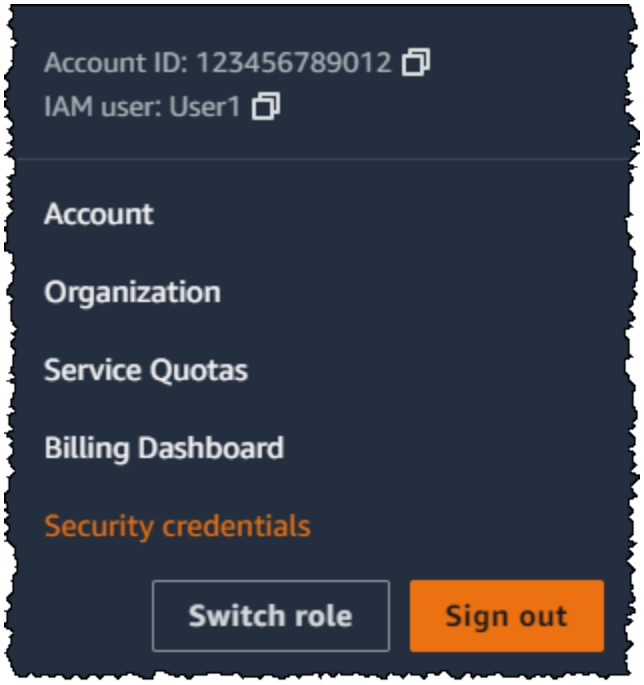
1. Nel pannello di navigazione seleziona Gruppi di utenti.
2. Nella casella di ricerca digitare **`EC2MFA`** e selezionare il nome del gruppo (non la casella di controllo) nell'elenco.
3. Nella scheda Permissions (Autorizzazioni), scegli Add permissions (Aggiungi autorizzazioni), quindi Attach policies (Collega policy).
4. Sulla pagina Collega policy di autorizzazione al gruppo EC2MFA nella casella di ricerca digita **`EC2Full`**. Quindi seleziona la casella di controllo accanto ad FullAccessAmazonEC2 nell'elenco. Non salvare ancora le modifiche.
5. Nella casella di ricerca, digitare **`Force`** e selezionare la casella di controllo accanto a Force_MFA nell'elenco.

6. Scegli Collega policy.

Fase 3: test dell'accesso dell'utente

In questa parte del tutorial, effettuare l'accesso come utente di prova e verificare che la policy funzioni correttamente.

1. Accedi al tuo annuncio **`MFAUser`** con Account AWS la password assegnata nella sezione precedente. Utilizzare l'URL: `https://<alias or account ID number>.signin.aws.amazon.com/console`
2. Seleziona EC2 per aprire la console Amazon EC2 e verifica che l'utente non disponga di autorizzazioni per effettuare alcuna operazione.
3. Selezionare il nome utente `MFAUser` in alto a destra nella barra di navigazione e scegli Security Credentials (Credenziali di sicurezza).



4. Aggiungere un dispositivo MFA. Nella sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori), scegliere Assign MFA device (Assegna dispositivo MFA).

> **Note**
>
> Potrebbe essere visualizzato un errore che indica che non si è autorizzati a eseguire `iam:DeleteVirtualMFADevice`. Questo può accadere se qualcuno in precedenza ha iniziato ad assegnare un dispositivo MFA virtuale a questo utente e ha annullato

il processo. Per continuare, l'utente o un altro amministratore devono eliminare il dispositivo MFA virtuale esistente non assegnato dell'utente. Per ulteriori informazioni, consulta Non sono autorizzato a eseguire: iam: MFADevice DeleteVirtual.

5. Per questo tutorial, utilizziamo un dispositivo MFA (basato su software), ad esempio Google Authenticator app su un cellulare. Scegli l'app Authenticator, quindi fai clic su Next (Successivo).

   IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della chiave di configurazione segreta che è disponibile per l'inserimento manuale su dispositivi che non supportano i codici QR.

6. Aprire l'app MFA virtuale. (Per un elenco di app che si possono utilizzare per ospitare dispositivi MFA virtuali, consulta Applicazioni MFA virtuali.) Se l'app MFA virtuale supporta più account (più dispositivi MFA virtuali), selezionare l'opzione che consente di creare un nuovo account (un nuovo dispositivo virtuale MFA).

7. Determinare se l'app MFA supporta i codici QR e procedere in uno dei seguenti modi:

   - Nella procedura guidata, scegliere Show QR code (Mostra codice QR). Quindi utilizzare l'app per la scansione del codice QR. Ad esempio, è possibile selezionare l'icona della fotocamera o un'opzione simile a Scannerizza codice ed eseguire la scansione del codice tramite la fotocamera del dispositivo.
   - Nella procedura guidata Set up device (Configura dispositivo), scegli Show secret key (Mostra chiave segreta) e digita la chiave segreta nell'app MFA.

   Al termine, il dispositivo MFA virtuale avvia la generazione di password una tantum.

8. Nella procedura guidata Set up device (Configura dispositivo), nella casella Enter the code from your authenticator app (Immetti il codice dall'app di autenticazione), digita la password una tantum che appare nel dispositivo MFA virtuale. Scegli Register MFA (Registra MFA).

   Important

   Invia la richiesta immediatamente dopo la generazione del codice. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA è correttamente associato all'utente. Tuttavia, il dispositivo MFA non è sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo MFA virtuale è ora pronto per l'uso con. AWS

9. Uscire dalla console ed effettuare nuovamente l'accesso come **`MFAUser`** Questa volta AWS ti viene richiesto un codice MFA dal tuo telefono. Una volta ottenuto, digitare il codice nella casella e selezionare Submit (Invia).
10. Seleziona EC2 per aprire nuovamente la console Amazon EC2. In questo momento è possibile visualizzare tutte le informazioni ed eseguire tutte le azioni desiderate. Se si accede a qualsiasi altra console come questo utente, vengono visualizzati messaggi di accesso negato. Il motivo è che le policy in questo tutorial concedono l'accesso solo a Amazon EC2.

Risorse correlate

Per ulteriori informazioni, consulta i seguenti argomenti:

- Utilizzo dell'autenticazione a più fattori (MFA) in AWS
- Abilitazione dei dispositivi MFA per gli utenti in AWS
- Utilizzo di dispositivi MFA con la pagina di accesso IAM

Identità IAM (utenti, gruppi di utenti e ruoli)

> **Tip**
>
> Hai problemi ad accedere a? AWS Assicurati di trovarti nella pagina di accesso corretta.
>
> - Per accedere come Utente root dell'account AWS (proprietario dell'account), usa le credenziali che hai impostato quando hai creato il Account AWS.
> - Per accedere come utente IAM, utilizza le credenziali fornite dall'amministratore dell'account per accedere ad AWS.
> - Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.
>
>   Per assistenza nell'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.
>
> Per i tutorial di accesso, consulta la pagina Come effettuare l'accesso ad AWS nella Guida per l'utente di Accedi ad AWS .

> **Note**
>
> Se hai bisogno di supporto, non utilizzare il link Feedback in questa pagina. Il feedback inserito viene ricevuto dal team di AWS documentazione, non dal AWS supporto. Scegli invece il link Contattaci nella parte superiore della pagina. Qui troverai i link alle risorse per aiutarti a ottenere il supporto di cui hai bisogno.

L'utente Utente root dell'account AWS o un utente amministrativo dell'account può creare identità IAM. Un'identità IAM consente l'accesso a un Account AWS. Un Gruppo di utenti IAM è una raccolta di utenti IAM gestiti come unità. Un'identità IAM rappresenta un utente umano o un carico di lavoro programmatico e può essere autenticato e quindi autorizzato a eseguire operazioni in AWS. Ogni identità IAM può essere associata a una o più policy. Le policy determinano quali azioni può eseguire un utente, un ruolo o un membro di un gruppo di utenti, su quali AWS risorse e in quali condizioni.

Account AWS utente root

La prima volta che si crea un account Account AWS, si inizia con un'unica identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità si chiama utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account.

> **Important**
>
> Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.

Utenti IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, le best practice raccomandano di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine. Se hai casi d'uso specifici che richiedono le chiavi di accesso, ti consigliamo di aggiornare le chiavi di accesso quando necessario. Per ulteriori informazioni, consulta Aggiornamento delle chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine. Per aggiungere utenti IAM al tuo Account AWS, consultaCreare un utente IAM nel tuo Account AWS.

> **Note**
>
> Come best practice di sicurezza, consigliamo di fornire l'accesso alle risorse tramite la federazione delle identità invece di creare utenti IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

Gruppi di utenti IAM

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile utilizzare un gruppo per effettuare l'accesso. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo chiamato IAMPublishers e assegnargli i tipi di autorizzazione dei quali hanno solitamente bisogno i carichi di lavoro dell'editoria.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo di ruoli, consulta Utilizzo di ruoli IAM.

I ruoli IAM con credenziali temporanee sono utilizzati nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, per alcuni dei Servizi AWS, è possibile collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulla differenza tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso alle risorse multi-account in IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua

applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
- Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Credenziali temporanee in IAM

Come best practice, utilizza credenziali temporanee sia per gli utenti che per i carichi di lavoro. Le credenziali temporanee vengono utilizzate principalmente con i ruoli IAM ma hanno anche altri usi. Puoi richiedere credenziali temporanee che hanno una serie di autorizzazioni più limitata rispetto

all'utente IAM standard. Ciò impedisce l'esecuzione accidentale di attività non consentite dalle credenziali più rigide. Un vantaggio delle credenziali temporanee è che scadono automaticamente dopo un determinato periodo di tempo. Puoi determinare la durata della validità delle credenziali.

In quali casi utilizzare utenti del Centro identità IAM?

Consigliamo a tutti gli utenti umani di utilizzare IAM Identity Center per accedere AWS alle risorse. IAM Identity Center consente miglioramenti significativi rispetto all'accesso alle AWS risorse come utente IAM. Il Centro identità IAM fornisce:

- Un insieme centrale di identità e assegnazioni
- Accesso agli account di un'intera AWS organizzazione
- Connessione al tuo gestore di identità esistente
- Credenziali temporanee
- Autenticazione a più fattori (MFA)
- Configurazione MFA self-service per utenti finali
- Applicazione amministrativa dell'uso dell'MFA
- Accesso unico a tutti i diritti Account AWS

Per ulteriori informazioni, consulta Cos'è il Centro di identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Quando creare un utente IAM invece di un ruolo

Ti consigliamo di utilizzare gli utenti IAM solo per casi d'uso non supportati dagli utenti federati. Alcuni dei casi d'uso sono i seguenti:

- Carichi di lavoro che non possono utilizzare ruoli IAM: è possibile eseguire un carico di lavoro da una posizione che deve accedere a AWS. In alcune situazioni, non puoi utilizzare i ruoli IAM per fornire credenziali temporanee, ad esempio per i plugin. WordPress In queste situazioni, per autenticarti a AWS usa le chiavi di accesso a lungo termine dell'utente IAM per quel carico di lavoro.
- AWS Client di terze parti: se utilizzi strumenti che non supportano l'accesso con IAM Identity Center, come AWS client o fornitori di terze parti che non sono ospitati su AWS, utilizza le chiavi di accesso a lungo termine degli utenti IAM.

- AWS CodeCommit accesso: se utilizzi CodeCommit per archiviare il codice, puoi utilizzare un utente IAM con chiavi SSH o credenziali specifiche del servizio CodeCommit per l'autenticazione nei tuoi repository. Si consiglia di eseguire questa operazione oltre a utilizzare un utente di IAM Identity Center per l'autenticazione normale. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue o alle tue applicazioni cloud. Account AWS Per consentire agli utenti di accedere ai tuoi CodeCommit repository senza configurare gli utenti IAM, puoi configurare l'utilità. git-remote-codecommit Per ulteriori informazioni su IAM e CodeCommit, consulta. Usare IAM con CodeCommit: credenziali Git, chiavi SSH e AWS chiavi di accesso Per ulteriori informazioni sulla configurazione dell'git-remote-codecommitutilità, consulta Connessione ai AWS CodeCommit repository con credenziali rotanti nella Guida per l'utente.AWS CodeCommit
- Accesso ad Amazon Keyspaces (per Apache Cassandra): in una situazione in cui non è possibile utilizzare gli utenti in IAM Identity Center, ad esempio per scopi di test per la compatibilità con Cassandra, puoi utilizzare un utente IAM con credenziali specifiche del servizio per l'autenticazione con Amazon Keyspaces. Gli utenti di IAM Identity Center sono le persone della tua forza lavoro che hanno bisogno di accedere alle tue applicazioni Account AWS o alle tue applicazioni cloud. Puoi anche connetterti ad Amazon Keyspaces utilizzando credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di credenziali temporanee per connettersi ad Amazon Keyspaces utilizzando un ruolo IAM e il plugin SIGv4 nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).
- Accesso di emergenza: in una situazione in cui non puoi accedere al tuo provider di identità e devi intervenire nel tuo Account AWS. Stabilire l'accesso di emergenza per gli utenti IAM può far parte del tuo piano di resilienza. Si consiglia di controllare e proteggere le credenziali degli utenti di emergenza con l'autenticazione a più fattori (MFA).

Quando creare un ruolo IAM invece di un utente

Crea un ruolo IAM nelle seguenti situazioni:

Stai creando un'applicazione che viene eseguita su un'istanza Amazon Elastic Compute Cloud (Amazon EC2) e tale applicazione invia richieste a. AWS

> Non creare un utente IAM e passare le credenziali dell'utente all'applicazione né integrare le credenziali nell'applicazione. Al contrario, crea un ruolo IAM da collegare all'istanza EC2 per fornire credenziali di sicurezza temporanee alle applicazioni in esecuzione sull'istanza. Quando un'applicazione utilizza queste credenziali in AWS, può eseguire tutte le operazioni consentite

dalle policy associate al ruolo. Per informazioni dettagliate, vedi Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Stai creando un'app che viene eseguita su un telefono cellulare e che effettua richieste ad AWS.

Non creare un utente IAM, né distribuire la chiave di accesso dell'utente con l'app. Al contrario, utilizza un provider di identità, come Amazon Cognito, Login with Amazon, Facebook o Google, per autenticare gli utenti e mapparli a un ruolo IAM. L'app può utilizzare il ruolo per ottenere credenziali di sicurezza temporanee con le autorizzazioni specificate dalle policy collegate al ruolo. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Guida per l'utente di Amazon Cognito
- Federazione OIDC

Gli utenti dell'azienda sono autenticati nella rete aziendale e desiderano poterli utilizzare AWS senza dover accedere nuovamente, ovvero consentire agli utenti di federarsi. AWS

Non creare utenti IAM. Configura una relazione di federazione tra il tuo sistema di identità aziendale e. AWS Ci sono due modi per farlo:

- Se il sistema di identità della tua azienda è compatibile con SAML 2.0, puoi stabilire un rapporto di fiducia tra il sistema di identità della tua azienda e AWS. Per ulteriori informazioni, consulta Federazione SAML 2.0.
- Crea e utilizza un server proxy personalizzato che traduce le identità degli utenti aziendali in ruoli IAM che forniscono credenziali di sicurezza temporanee AWS . Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

Confronta le Utente root dell'account AWS credenziali e le credenziali utente IAM

L'utente root è il proprietario dell'account e viene creato al momento della Account AWS creazione di. Altri tipi di utenti, inclusi gli utenti IAM, e AWS IAM Identity Center gli utenti vengono creati dall'utente root o da un amministratore dell'account. Tutti AWS gli utenti dispongono di credenziali di sicurezza.

Credenziali utente root

Le credenziali del proprietario dell'account consentono il pieno accesso a tutte le risorse nell'account. Non è possibile utilizzare policy IAM per negare esplicitamente all'utente root l'accesso alle risorse. È possibile utilizzare solo una policy AWS Organizations di controllo del servizio (SCP) per limitare

le autorizzazioni dell'utente root di un account membro. Per questo motivo, ti consigliamo di creare un utente amministrativo in IAM Identity Center da utilizzare per le attività quotidiane AWS . Quindi, proteggi le credenziali dell'utente root e utilizzale per eseguire solo quelle poche attività di gestione di account e servizi che richiedono l'accesso come utente root. Per visualizzare un elenco di queste attività, consulta la pagina Attività che richiedono credenziali dell'utente root. Per scoprire come configurare un amministratore per l'uso quotidiano in Centro identità IAM, consulta la pagina Nozioni di base della Guida per l'utente di Centro identità IAM.

Credenziali IAM

Un utente IAM è un'entità in cui crei AWS che rappresenta la persona o il servizio che utilizza l'utente IAM per interagire con AWS le risorse. Questi utenti sono identità interne all'utente Account AWS che dispongono di autorizzazioni personalizzate specifiche. Ad esempio, è possibile creare utenti IAM e concedere loro le autorizzazioni per creare una directory in Centro identità IAM. Gli utenti IAM dispongono di credenziali a lungo termine che possono utilizzare per accedere AWS utilizzando le o AWS Management Console API o a livello di codice. AWS CLI AWS Per step-by-step istruzioni su come gli utenti IAM accedono a AWS Management Console, consulta Accedere AWS Management Console come utente IAM nella Guida per l'utente di accesso.AWS

In generale, ti consigliamo di evitare la creazione di utenti IAM perché dispongono di credenziali a lungo termine, ad esempio nome utente e password. Richiedi invece agli utenti umani di utilizzare credenziali temporanee durante l'accesso. AWS Puoi utilizzare un provider di identità a cui gli utenti umani possano fornire un accesso federato Account AWS assumendo ruoli IAM, che forniscono credenziali temporanee. Per una gestione centralizzata degli accessi, consigliamo di utilizzare Centro identità IAM per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Puoi gestire le tue identità utente con IAM Identity Center o gestire le autorizzazioni di accesso per le identità degli utenti in IAM Identity Center da un provider di identità esterno. Per ulteriori informazioni su Centro identità IAM, consulta la pagina Cos'è Centro identità IAM? nella Guida per l'utente di Centro identità IAM.

Utente root dell'account AWS

Quando crei per la prima volta un account Amazon Web Services (AWS), inizi con un'identità di accesso singolo che ha accesso completo a tutti i AWS servizi e le risorse dell'account. Questa identità si chiama utente root dell' AWS account ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account.

> **Important**
>
> Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane e di seguire le best practice dell'utente root per il tuo Account AWS. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.

I seguenti argomenti descrivono in dettaglio le attività di gestione associate all'utente root.

Attività

- Abilita l'autenticazione a più fattori per la tua Utente root dell'account AWS (console)
- Cambiare la password per Utente root dell'account AWS
- Reimpostazione di una password dell'utente root persa o dimenticata
- Creazione di chiavi di accesso per l'utente root
- Eliminazione di chiavi di accesso per l'utente root
- Attività che richiedono credenziali dell'utente root
- Risoluzione dei problemi dell'utente root
- Informazioni correlate

Abilita l'autenticazione a più fattori per la tua Utente root dell'account AWS (console)

L'autenticazione a più fattori (MFA) è un meccanismo semplice ed efficace per migliorare la sicurezza. Il primo fattore, la password, è un segreto che memorizzi, noto anche come fattore di conoscenza. Altri fattori possono essere fattori di possesso (qualcosa che possiedi, come una chiave di sicurezza) o fattori intrinseci (qualcosa che sei, come una scansione biometrica). Per una maggiore sicurezza, ti consigliamo vivamente di configurare l'autenticazione a più fattori (MFA) per proteggere AWS le tue risorse.

Puoi abilitare l'MFA per gli utenti Utente root dell'account AWS e IAM. Quando si abilita l'MFA per l'utente root, ciò influisce solo sulle credenziali dell'utente root. Per ulteriori informazioni su come abilitare l'MFA per gli utenti IAM, consulta Abilitazione dei dispositivi MFA per gli utenti IAM in. AWS

Prima di abilitare la MFA per l'utente root, rivedi e aggiorna le impostazioni dell'account e le informazioni di contatto per assicurarti di avere accesso all'e-mail e al numero di telefono. Se il dispositivo MFA viene smarrito, rubato o non funziona, è comunque possibile accedere come utente root verificando la propria identità utilizzando tale e-mail e il numero di telefono. Per ulteriori informazioni sull'accesso utilizzando fattori di autenticazione alternativi, consultare Cosa fare se un dispositivo MFA viene smarrito o smette di funzionare?. Per disabilitare questa funzionalità, contatta AWS Support.

Argomenti

Tipi di MFA disponibili per un utente root

AWS supporta i seguenti tipi di MFA per l'utente root: passkey e chiavi di sicurezza, applicazioni di autenticazione virtuale e token TOTP hardware.

Passkey e chiavi di sicurezza

AWS Identity and Access Management supporta passkey e chiavi di sicurezza per MFA. In base agli standard FIDO, le passkey utilizzano la crittografia a chiave pubblica per fornire un'autenticazione forte e resistente al phishing, più sicura delle password. AWS supporta due tipi di passkey: passkey legate al dispositivo (chiavi di sicurezza) e passkey sincronizzate.

- Chiavi di sicurezza: si tratta di dispositivi fisici, come un YubiKey, utilizzati come secondo fattore di autenticazione.
- Passkey sincronizzate: come secondo fattore utilizzano gestori di credenziali di provider come Google, Apple, account Microsoft e servizi di terze parti come 1Password, Dashlane e Bitwarden.

Puoi utilizzare gli autenticatori biometrici integrati, come Touch ID su Apple MacBooks e il riconoscimento facciale Windows Hello sui PC, per sbloccare il gestore delle credenziali e accedere a. AWS Le passkey vengono create con il provider prescelto utilizzando il PIN dell'impronta digitale, del viso o del dispositivo. Puoi sincronizzare le passkey tra i tuoi dispositivi per facilitare gli accessi e migliorare l'usabilità e la AWS recuperabilità.

FIDO Alliance mantiene un elenco di tutti i prodotti certificati FIDO compatibili con le specifiche FIDO. Una singola passkey o chiave di sicurezza supporta più account utente root e utenti IAM. Per ulteriori informazioni sull'abilitazione delle passkey e delle chiavi di sicurezza, consulta. Abilita una passkey o una chiave di sicurezza per Utente root dell'account AWS (console)

Applicazioni di autenticazione virtuale

Un'applicazione di autenticazione virtuale viene eseguita su un telefono o un altro dispositivo ed emula un dispositivo fisico. Le app di autenticazione virtuale implementano l'algoritmo TOTP (password monouso) e supportano più token su un singolo dispositivo. L'utente deve digitare un codice valido dal dispositivo quando richiesto durante l'accesso. Ogni token assegnato a un utente deve essere unico. Un utente non può digitare un codice dal token di un altro utente per l'autenticazione.

È consigliabile utilizzare un dispositivo MFA virtuale nell'attesa dell'approvazione di un acquisto hardware o della consegna del dispositivo hardware. Per un elenco di alcune app supportate che puoi utilizzare come dispositivi MFA virtuali, consulta Multi-Factor Authentication (MFA). Per istruzioni sulla configurazione di un dispositivo MFA virtuale con AWS, vedere. Abilitazione di un dispositivo MFA virtuale per l' Utente root dell'account AWS (console)

Token TOTP hardware

Un dispositivo hardware genera un codice numerico a sei cifre basato sull'algoritmo TOTP (Time-based One-Time Password). L'utente deve immettere un codice valido dal dispositivo su una seconda pagina Web durante la procedura di accesso. Ogni dispositivo MFA assegnato a un utente deve essere univoco. Per essere autenticati, gli utenti non possono digitare un codice generato dal dispositivo di un altro utente. Per informazioni sui dispositivi hardware MFA supportati, vedere Multi-Factor Authentication (MFA). Per istruzioni sulla configurazione di un token TOTP hardware con, consulta. AWSAbilita un token TOTP hardware per Utente root dell'account AWS (console)

Se desideri utilizzare un dispositivo MFA fisico, ti consigliamo di utilizzare le chiavi di sicurezza FIDO come alternativa ai dispositivi TOTP hardware. Le chiavi di sicurezza FIDO offrono i vantaggi di non richiedere alcuna batteria, resistono al phishing e supportano più utenti root e IAM su un unico dispositivo per una maggiore sicurezza.

Abilita una passkey o una chiave di sicurezza per Utente root dell'account AWS (console)

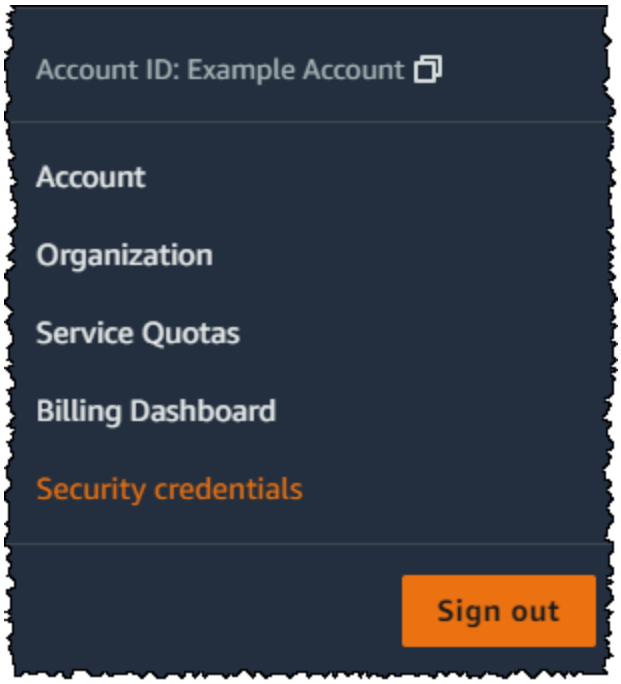
È possibile configurare e abilitare una passkey per l'utente root AWS Management Console solo dall'API, non dall'API AWS CLI . AWS

Per abilitare una passkey o una chiave di sicurezza per l'utente root (console)

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Scegli il nome dell'account sul lato destro della barra di navigazione, quindi seleziona Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Nella pagina Le mie credenziali di sicurezza dell'utente root, in Autenticazione a più fattori (MFA), scegli Assegna dispositivo MFA.
4. Nella pagina del nome del dispositivo MFA, inserisci un nome dispositivo, scegli Passkey o Security Key, quindi scegli Avanti.

5. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.
6. Segui le istruzioni del tuo browser per scegliere un fornitore di passkey o dove vuoi archiviare la passkey da utilizzare su tutti i tuoi dispositivi.
7. Scegli Continua.

Ora hai registrato la tua passkey per utilizzarla con. AWS La prossima volta che utilizzerai le credenziali dell'utente root per accedere, dovrai autenticarti con la tua passkey per completare la procedura di accesso.

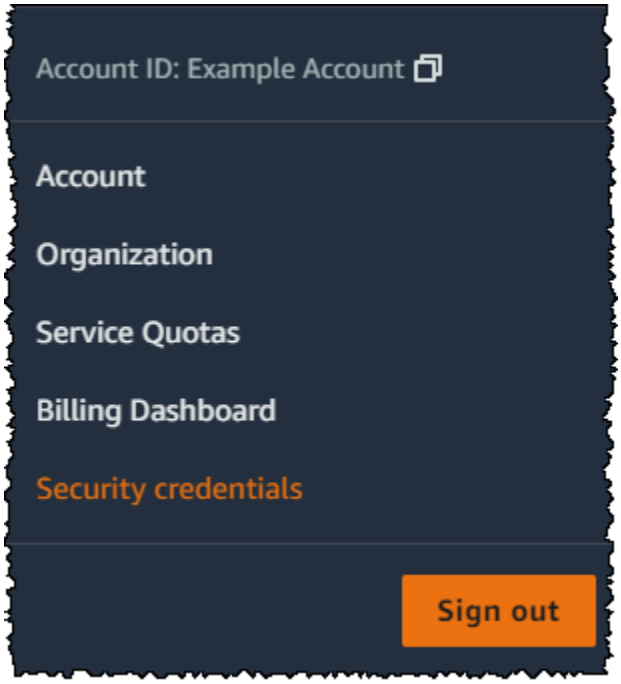
Per assistenza nella risoluzione dei problemi relativi alla chiave di sicurezza FIDO, consulta. Risoluzione dei problemi relativi alle chiavi di sicurezza FIDO

Abilitazione di un dispositivo MFA virtuale per l' Utente root dell'account AWS (console)

È possibile utilizzare il AWS Management Console per configurare e abilitare un dispositivo MFA virtuale per l'utente root. Per abilitare i dispositivi MFA per Account AWS, è necessario accedere AWS utilizzando le credenziali dell'utente root.

Come configurare e abilitare un dispositivo MFA virtuale da utilizzare con l'utente root (console)

1. Accedi alla AWS Management Console.
2. Selezionare il nome dell'account sul lato destro della barra di navigazione e scegliere Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security credentials (Continua con le credenziali di sicurezza).



3. Nella sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori), scegliere Assign MFA device (Assegna dispositivo MFA).
4. Nella procedura guidata, digita un nome per il dispositivo, scegli l'app Authenticator e quindi scegli Next (Avanti).

   IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della chiave di configurazione segreta che è disponibile per l'inserimento manuale su dispositivi che non supportano i codici QR.
5. Aprire l'app MFA virtuale sul dispositivo.

   Se l'app MFA virtuale supporta più account o dispositivi MFA virtuali, selezionare l'opzione che consente di creare un nuovo account o dispositivo MFA virtuale.
6. Il modo più semplice per configurare l'app è di utilizzare l'app per scannerizzare il codice QR. Se non è possibile scansionare il codice, è possibile digitare le informazioni di configurazione manualmente. Il codice QR e la chiave di configurazione segreta generati da IAM sono collegati al tuo Account AWS e non possono essere utilizzati con un altro account. Possono tuttavia essere riutilizzati per configurare un nuovo dispositivo MFA per l'account nel caso in cui si perda l'accesso al dispositivo MFA originale.

   - Per utilizzare il codice QR per la configurazione del dispositivo MFA virtuale, scegliere Show QR code (Mostra codice QR) nella procedura guidata. Quindi, seguire le istruzioni nell'app relative alla scansione del codice. Ad esempio, potrebbe essere necessario scegliere l'icona

della fotocamera o un comando come Scan account barcode (Scannerizza codice a barre account) e utilizzare la fotocamera del dispositivo per eseguire la scansione del codice QR.

- Nella procedura guidata Set up device (Configura dispositivo), scegli Show secret key (Mostra chiave segreta) e digita la chiave segreta nell'app MFA.

> **Important**
>
> Effettuare un backup sicuro del codice QR o della chiave segreta di configurazione o assicurarsi che più dispositivi MFA virtuali siano abilitati per il proprio account. Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Un dispositivo MFA virtuale potrebbe non essere più disponibili, ad esempio, se si perde lo smartphone, in cui il dispositivo MFA virtuale è ospitato. In tal caso e se non riesci ad accedere al tuo account senza dispositivi MFA aggiuntivi collegati all'utente né tramite Ripristino di un dispositivo MFA per l'utente root, non potrai accedere al tuo account e dovrai contattare l'assistenza clienti per rimuovere la protezione MFA per l'account.

Il dispositivo avvia la generazione di numeri a sei cifre.

7. Nella procedura guidata, nella casella MFA Code 1 (Codice MFA 1), digita la password monouso visualizzata nel dispositivo MFA virtuale. Attendere fino a un massimo di 30 secondi prima che il dispositivo generi una nuova password una tantum. Quindi, digitare la seconda password monouso nella casella Codice MFA 2. Scegli Aggiungi MFA.

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA si associa correttamente con l'utente ma il dispositivo MFA non viene sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con. AWS Per ulteriori informazioni sull'utilizzo di MFA con la AWS Management Console, consulta Utilizzo di dispositivi MFA con la pagina di accesso IAM.

Abilita un token TOTP hardware per Utente root dell'account AWS (console)

Puoi configurare e abilitare un dispositivo MFA fisico per il tuo utente root AWS Management Console solo dall'API, non dall'API AWS CLI o AWS .

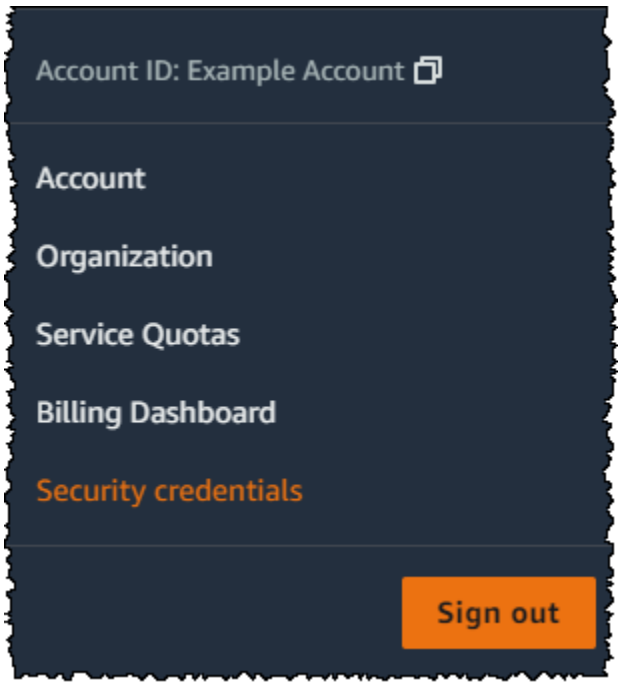
> **Note**
>
> È possibile che il testo visualizzato sia differente, ad esempio Sign in using MFA (Accesso con un dispositivo MFA) e Troubleshoot your authentication device (Risoluzione dei problemi del dispositivo di autenticazione). Tuttavia, le funzionalità sono identiche. In entrambi i casi, se non fosse possibile verificare l'indirizzo e-mail e il numero di telefono dell'account utilizzando fattori alternativi dell'autenticazione, contatta AWS Support per disattivare l'impostazione MFA.

Come abilitare il dispositivo MFA per l'utente root (console)

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Sul lato destro della barra di navigazione, seleziona il nome dell'account, quindi Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Espandere la sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori (MFA)).
4. Scegli Assegna dispositivo MFA.
5. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
6. Nella casella Serial number (Numero di serie) digitare il numero di serie riportato sulla parte posteriore del dispositivo MFA.
7. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.

8. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
9. Scegli Aggiungi MFA. Il dispositivo MFA è ora associato all'account Account AWS.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Al successivo utilizzo delle credenziali dell'utente root per effettuare l'accesso, dovrai immettere un codice dal dispositivo MFA.

Cambiare la password per Utente root dell'account AWS

Puoi modificare l'indirizzo e-mail e la password in Credenziali di sicurezza o nella pagina Account. Puoi anche scegliere Password dimenticata? nella pagina di AWS accesso per reimpostare la password.

Per modificare la password dell'utente root, devi accedere come utente IAM Utente root dell'account AWS e non come utente. Per ulteriori informazioni su come reimpostare una password dell'utente root dimenticata, consulta Reimpostazione di una password dell'utente root persa o dimenticata.

Per proteggere la password, consigliamo di seguire queste best practice:

- Cambia periodicamente la password.
- Mantieni la password privata, perché chiunque la conosca può accedere al tuo account.
- Usa una password diversa da AWS quella che usi su altri siti.
- Evitare password che sono facili da indovinare. Queste includono password, come `secret`, `password`, `amazon` o `123456`. Sono inclusi anche i dati come una parole comuni, il tuo nome, l'indirizzo e-mail o altre informazioni personali che possono essere ottenute facilmente.

AWS Management Console

Come modificare la password per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Usa Account AWS il tuo indirizzo e-mail e la password per accedere AWS Management Consolecome tuoi Utente root dell'account AWS.
2. Nell'angolo in alto a destra della console, scegli il nome o il numero dell'account, quindi Account.
3. Nella pagina Account, accanto a AImpostazioni account, scegli Modifica. Ti viene richiesto di effettuare nuovamente l'autenticazione per motivi di sicurezza.

   > **Note**
   >
   > Se non vedi l'opzione Modifica, è probabile che tu non abbia effettuato l'accesso come utente root dell'account. Non è possibile modificare le impostazioni dell'account dopo aver effettuato l'accesso come utente o ruolo IAM.

4. Nella pagina Aggiorna le impostazioni dell'account, in Password, scegli Modifica.
5. Nella pagina Aggiorna la password, compila i campi Password corrente, Nuova password e Conferma nuova password.

   > **Important**
   >
   > Scegli una password sicura. Anche se è possibile impostare una policy per le password dell'account per gli utenti IAM, tale policy non si applica all'utente root.

   AWS richiede che la password soddisfi le seguenti condizioni:

- Deve avere un minimo di 8 caratteri e un massimo di 128 caratteri.
- Deve includere almeno tre dei seguenti tipi di caratteri: maiuscole, minuscole, numeri e i simboli ! @ # $ % ^ & * () <> [] {} | _+-=.
- Non deve essere identica al tuo Account AWS nome o indirizzo email.

> **Note**
>
> AWS sta introducendo miglioramenti alla procedura di accesso. Uno di questi miglioramenti è quello di implementare una policy delle password più sicure per l'account. Se AWS ha effettuato l'upgrade dell'account, è necessario soddisfare la policy sulla password descritta sopra. Se AWS non ha ancora effettuato l'upgrade del tuo account, AWS significa che non applica ancora questa politica. Tuttavia, consigliamo fortemente di seguire le predette linee guida per incrementare la sicurezza della password.

6. Seleziona Salvataggio delle modifiche.

AWS CLI or AWS SDK

Questa attività non è supportata in AWS CLI o da un'operazione API di uno degli AWS SDK. È possibile eseguire questa attività solo utilizzando. AWS Management Console

Reimpostazione di una password dell'utente root persa o dimenticata

La prima volta che hai creato il tuo Account AWS, hai fornito un indirizzo email e una password. Queste sono le tue Utente root dell'account AWS credenziali. Se dimentichi la password dell'utente root, puoi reimpostarla dalla AWS Management Console.

Per reimpostare la password dell'utente root:

1. Usa il tuo indirizzo Account AWS e-mail per iniziare ad accedere AWS Management Consolecome utente root, quindi scegli Avanti.

> **Note**
>
> Se è stato effettuato l'accesso alla AWS Management Console con le credenziali dell'utente IAM, per reimpostare la password dell'utente root è necessario prima disconnettersi. Se viene visualizzata la pagina di accesso dell'utente IAM specifica dell'account, seleziona Accedi con le credenziali dell'account root nella parte inferiore della pagina. Se necessario, fornisci l'indirizzo e-mail dell'account e scegli Next (Avanti) per accedere alla pagina Root user sign in (Accesso utente root).

2. Selezionare Forgot your password? (Password dimenticata?).

> **Note**
>
> Se sei un utente IAM, questa opzione non è disponibile. L'opzione Password dimenticata? è disponibile solo per l'account utente root. Gli utenti IAM devono chiedere al proprio amministratore di reimpostare una password dimenticata. Per ulteriori informazioni, consulta Ho dimenticato la password utente IAM per il mio AWS account. Se accedi tramite Portale di accesso AWS, consulta Reimpostazione della password utente di IAM Identity Center.

3. Fornire l'indirizzo e-mail associato all'account. Fornire quindi il testo CAPTCHA e selezionare Continue (Continua).
4. Verifica la presenza di un messaggio proveniente da Amazon Web Services nell'indirizzo e-mail associato al tuo Account AWS . L'e-mail proviene da un indirizzo che termina con `@verify.signin.aws`. Seguire le istruzioni nel messaggio. Se il messaggio e-mail non viene visualizzato nel proprio account, controllare la cartella spam. Se non hai più accesso all'e-mail, consulta Non ho accesso all'e-mail del mio AWS account nella Guida per l'Accedi ad AWS utente.

Creazione di chiavi di accesso per l'utente root

> **Warning**
>
> Consigliamo fortemente di non creare coppie di chiavi di accesso per l'utente root. Poiché solo alcune attività richiedono l'utente root e in genere tali attività vengono eseguite raramente, si consiglia di accedere a per eseguire le AWS Management Console attività

dell'utente root. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Sebbene non sia consigliabile, puoi creare chiavi di accesso per il tuo utente root in modo da poter eseguire i comandi in AWS Command Line Interface (AWS CLI) o utilizzare le operazioni API da uno degli AWS SDK utilizzando le credenziali dell'utente root. Quando crei una chiave di accesso, crei l'ID chiave di accesso e la chiave di accesso segreta come set. Durante la creazione della chiave di accesso, ti AWS offre l'opportunità di visualizzare e scaricare la parte della chiave di accesso segreta della chiave di accesso. Se non la scarichi o la perdi, puoi eliminare la chiave di accesso e quindi crearne una nuova. È possibile creare chiavi di accesso per utenti root con la AWS CLI console o AWS l'API.

Lo stato di una nuova chiave di accesso è attiva, il che significa che puoi usarla per le chiamate API e CLI. Inoltre, è possibile assegnare fino a due chiavi di accesso all'utente root.

Le chiavi di accesso non utilizzate dovrebbero essere disattivate. Una volta che una chiave di accesso è inattiva, non è possibile utilizzarla per le chiamate API. Le chiavi inattive contano comunque per il limite. Puoi creare o eliminare una chiave di accesso in qualsiasi momento. Tuttavia, una volta eliminata, viene persa per sempre e non può essere recuperata.

AWS Management Console

Per creare una chiave di accesso per Utente root dell'account AWS

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, operazione che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Usa l'indirizzo email e la password Account AWS del tuo account per accedere alla Guida introduttiva usando AWS Management Console come tuoi Utente root dell'account AWS.
2. Nell'angolo in alto a destra della console, seleziona il nome o il numero dell'account, quindi scegli Credenziali di sicurezza.

3. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Se questa opzione non è disponibile, è già stato raggiunto il numero massimo di chiavi di accesso. È necessario eliminare una delle chiavi di accesso esistenti prima di poter creare una nuova chiave. Per ulteriori informazioni, consulta la pagina Quote degli oggetti IAM.
4. Nella pagina Alternative alle chiavi di accesso dell'utente root, consulta le raccomandazioni di sicurezza. Per continuare, seleziona la casella di controllo, quindi scegli Crea chiave di accesso.
5. Nella pagina Recupera la chiave d'accesso, viene visualizzato l'ID della Chiave di accesso.
6. In Chiave di accesso segreta, seleziona Mostra, quindi copia l'ID della chiave di accesso e della chiave segreta dalla finestra del browser e incollale in un luogo sicuro. In alternativa, puoi selezionare Scarica file .csv: eseguirai il download di un file denominato `rootkey.csv` che contiene l'ID chiave di accesso e la chiave segreta. Salvare il file da qualche parte al sicuro.
7. Seleziona Fatto. Quando non hai più bisogno della chiave di accesso, ti consigliamo di eliminarla o almeno di valutare se disattivarla, in modo che nessuno possa usarla in modo improprio.

AWS CLI & SDKs

Per creare una chiave di accesso per l'utente root

> **Note**
>
> Per eseguire il seguente comando od operazione API come utente root, è necessario disporre già di una coppia di chiavi di accesso attive. Se non disponi delle chiavi di accesso, crea la prima chiave di accesso utilizzando la AWS Management Console. Quindi, puoi utilizzare le credenziali della prima chiave di accesso con le AWS CLI per creare la seconda chiave di accesso o per eliminare una chiave di accesso.

- AWS CLI: era io create-access-key

  Example

```
$ aws iam create-access-key
{
    "AccessKey": {
```

```
        "UserName": "MyUserName",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "Status": "Active",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "CreateDate": "2021-04-08T19:30:16+00:00"
    }
}
```

- AWS API: CreateAccessKeynello IAM API Reference.

Eliminazione di chiavi di accesso per l'utente root

È possibile utilizzare l' AWS Management Console, the AWS CLI o l' AWS API per eliminare le chiavi di accesso dell'utente root.

AWS Management Console

Per eliminare una chiave di accesso per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, il che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

1. Usa l'indirizzo email e la password Account AWS del tuo account per accedere alla Guida introduttiva usando AWS Management Console come tuoi Utente root dell'account AWS.
2. Nell'angolo in alto a destra della console, seleziona il nome o il numero dell'account, quindi scegli Credenziali di sicurezza.
3. Nella sezione Chiavi di accesso, individua la chiave di accesso che desideri eliminare, quindi scegli Operazioni e poi Elimina.

> **Note**
>
> In alternativa, puoi disattivare una chiave di accesso anziché eliminarla definitivamente. In questo modo potrai riutilizzarla in futuro senza dover modificare l'ID chiave o la chiave segreta. Sebbene la chiave sia inattiva, qualsiasi tentativo di utilizzarla nelle richieste all' AWS API fallisce e viene negato l'errore di accesso.

4. Nella finestra di dialogo Elimina <ID chiave di accesso>, scegli Disattiva, inserisci l'ID della chiave di accesso per confermare che desideri eliminarla, quindi scegli Elimina.

AWS CLI & SDKs

Per eliminare una chiave di accesso per l'utente root

> **Autorizzazioni minime**
>
> Per eseguire le seguenti operazioni, devi disporre come minimo delle seguenti autorizzazioni IAM:
>
> - È necessario accedere come utente Account AWS root, operazione che non richiede autorizzazioni aggiuntive AWS Identity and Access Management (IAM). Non è possibile eseguire questi passaggi come utente o ruolo IAM.

- AWS CLI: era iam delete-access-key

  Example

  ```
  $ aws iam delete-access-key \
      --access-key-id AKIAIOSFODNN7EXAMPLE
  ```

  Questo comando non produce alcun output se ha esito positivo.

- AWS API: DeleteAccessKey

Attività che richiedono credenziali dell'utente root

> **Important**
>
> Hai problemi ad accedere a AWS? Assicurati di essere nella pagina di accesso AWS corretta per il tuo tipo di utente. Se sei il Utente root dell'account AWS (proprietario dell'account), puoi accedere AWS utilizzando le credenziali che hai configurato quando hai creato il Account AWS. Se sei un utente IAM, l'amministratore dell'account può fornirti le credenziali che puoi utilizzare per accedere ad AWS. Se hai bisogno di richiedere assistenza, non utilizzare il link di feedback in questa pagina, poiché il modulo non AWS Support viene ricevuto dal team addetto alla AWS documentazione. Invece, nella pagina Contattaci scegli Ancora impossibile accedere al tuo AWS account, quindi scegli una delle opzioni di supporto disponibili.

Ti consigliamo di configurare un utente amministrativo AWS IAM Identity Center per eseguire le attività quotidiane e accedere alle AWS risorse. Tuttavia, è possibile eseguire le attività elencate di seguito solo se si effettua l'accesso come utente root di un account.

Attività di gestione degli account

- Cambiare le impostazioni dell'account. Sono inclusi il nome dell'account, l'indirizzo e-mail, la password dell'utente root e le chiavi di accesso dell'utente root. Altre impostazioni dell'account, come le informazioni di contatto, la preferenza per la valuta di pagamento e Regioni AWS, non richiedono credenziali dell'utente root.
- Ripristina le autorizzazioni dell'utente IAM. Se l'unico amministratore IAM revoca accidentalmente le autorizzazioni, sarà possibile effettuare l'accesso come utente root per modificare le policy e ripristinare le autorizzazioni.
- Chiudi il tuo. Account AWS

  Per ulteriori informazioni, consulta i seguenti argomenti:

  - Come posso assegnare la proprietà del mio Account AWS a un'altra entità? .
  - Come faccio a chiudere il mio Account AWS? .
  - Chiudi uno standalone. Account AWS

Attività di fatturazione

- Attivare l'accesso IAM alla console di gestione fatturazione e costi.

- Alcune attività di fatturazione sono limitate all'utente root. Per ulteriori informazioni, consulta Managing an Account AWS in AWS Billing User Guide.
- Visualizzare alcune fatture fiscali. Un utente IAM con il portale aws: l'ViewBillingautorizzazione può visualizzare e scaricare le fatture IVA dall' AWS Europa, ma non da AWS Inc. o Amazon Internet Services Private Limited (AISPL).

AWS GovCloud (US) Attività

- Registrazione per AWS GovCloud (US).
- Richiedi le chiavi di accesso per l'utente root dell' AWS GovCloud (US) account da AWS Support.

Attività Amazon EC2

- Registrati come venditore nel marketplace di istanze riservate.

AWS KMS Attività

- Nel caso in cui una AWS Key Management Service chiave diventi ingestibile, un amministratore può recuperarla contattandola AWS Support; tuttavia, AWS Support risponde al numero di telefono principale dell'utente root per l'autorizzazione confermando l'OTP del ticket.

Attività di Amazon Mechanical Turk

- Collega il tuo Account AWS al tuo account MTurk Requester.

Attività di Amazon Simple Storage Service

- Configura un bucket Amazon S3 per abilitare l'autenticazione a più fattori (MFA).
- Modifica o elimina una policy sui bucket di Amazon S3 che nega tutti i principi.

Attività del servizio Amazon Simple Queue

- Modifica o elimina una policy sulle risorse di Amazon SQS che nega tutti i principali.

Risoluzione dei problemi dell'utente root

Le informazioni seguenti ti aiutano a risolvere i problemi relativi all'utente root di un Account AWS.

Non riesco a eseguire le attività che mi aspetto di poter eseguire quando effettuo l'accesso come utente root dell'account

Se non riesci a completare le attività quando hai eseguito l'accesso come utente root dell'account, l'account potrebbe essere membro di un'organizzazione in AWS Organizations. In tal caso, e se l'amministratore dell'organizzazione ha utilizzato una policy di controllo dei servizi per limitare le autorizzazioni dell'account, sono interessati tutti gli utenti, incluso l'utente root. Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .

Ho dimenticato la password dell'utente root per il mio Account AWS

Se sei un utente root e hai perso o dimenticato la tua password Account AWS, puoi reimpostarla. È necessario conoscere l'indirizzo e-mail utilizzato per creare l' Account AWS e disporre dell'accesso all'account e-mail. Per ulteriori informazioni, consulta Reimpostazione di una password dell'utente root persa o dimenticata.

Non ho accesso all'e-mail del mio Account AWS

Quando ne crei uno Account AWS, fornisci un indirizzo email e una password. Queste sono le credenziali per Utente root dell'account AWS. Se non sei sicuro dell'indirizzo e-mail associato al tuo Account AWS, cerca i messaggi inviati da `@signin.aws` o `@verify.signin.aws` verso qualsiasi indirizzo e-mail della tua organizzazione che potrebbe essere stato utilizzato per aprire il Account AWS.

Se conosci l'indirizzo e-mail ma non hai più accesso all'e-mail, prova innanzitutto a recuperare l'accesso all'e-mail utilizzando una delle seguenti opzioni:

- Se sei il proprietario del dominio dell'indirizzo e-mail, puoi ripristinare un indirizzo e-mail eliminato. In alternativa, puoi impostare un catch-all per il tuo account e-mail che "acquisisce tutti" i messaggi inviati a indirizzi e-mail che non esistono più nel server di posta e li reindirizza a un altro indirizzo e-mail.
- Se l'indirizzo e-mail dell'account è parte del sistema di posta elettronica aziendale, si consiglia di contattare gli amministratori del sistema IT. Potrebbero essere in grado di aiutare a ottenere nuovamente l'accesso all'e-mail.

Se ancora non riesci ad accedere al tuo Account AWS, puoi trovare opzioni di supporto alternative alla pagina Contattaci.

Informazioni correlate

I seguenti articoli forniscono ulteriori informazioni sull'utilizzo dell'utente root.

- Quali sono le migliori pratiche per proteggere le mie Account AWS e le sue risorse?
- Come posso creare una regola di EventBridge evento per informarmi che è stato utilizzato il mio utente root?
- Monitora e invia notifiche sulle Utente root dell'account AWS attività
- Monitoraggio dell'attività dell'utente root IAM

Utenti IAM

> **Important**
>
> Le migliori pratiche IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché utilizzare utenti IAM con credenziali a lungo termine.

Un utente AWS Identity and Access Management (IAM) è un'entità in cui crei. AWS L'utente IAM rappresenta l'utente o il carico di lavoro umano che utilizza l'utente IAM con AWS cui interagire. Un utente in AWS è composto da un nome e da credenziali.

Un utente IAM con le autorizzazioni di amministratore non è la stessa cosa dell' Utente root dell'account AWS. Per ulteriori informazioni sull'utilizzo dell'utente root, consulta Utente root dell'account AWS.

Come AWS identifica un utente IAM

Quando crei un utente IAM, IAM crea questi metodi per identificare quell'utente:

- Un "nome semplice" per l'utente IAM, che è il nome specificato quando hai creato l'utente IAM, ad esempio `Richard` o `Anaya`. Questi sono i nomi che vedi nella AWS Management Console.
- Un nome della risorsa Amazon (ARN) per l'utente IAM. L'ARN viene utilizzato quando è necessario identificare in modo univoco l'utente IAM in tutto il mondo. AWS Ad esempio, puoi usare un ARN

per specificare l'utente IAM come `Principal` in una policy IAM per un bucket Amazon S3. Un ARN per un utente IAM potrebbe essere simile al seguente:

`arn:aws:iam::account-ID-without-hyphens:user/Richard`

- Un identificatore univoco per l'utente IAM. Questo ID viene restituito solo quando utilizzi l'API, Tools for Windows PowerShell o AWS CLI per creare l'utente IAM; non lo vedi nella console.

Per ulteriori informazioni su questi identificatori, consulta Identificatori IAM.

Utenti IAM e credenziali

Puoi accedere AWS in diversi modi a seconda delle credenziali utente IAM:

- Password console: una password che l'utente IAM può inserire per accedere a sessioni interattive come la AWS Management Console. La disabilitazione della password (accesso alla console) per un utente IAM impedisce loro di accedere AWS Management Console utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto.
- Tasti di accesso: utilizzati per effettuare chiamate programmatiche a AWS. Tuttavia, ci sono alternative più sicure da considerare prima di creare le chiavi di accesso per gli utenti IAM. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella Riferimenti generali di AWS. Se l'utente IAM dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l' AWS CLI accesso tramite Tools for Windows PowerShell, AWS API o l'applicazione AWS Console Mobile.
- Chiavi SSH da usare con CodeCommit: una chiave pubblica SSH in formato OpenSSH che può essere utilizzata per l'autenticazione. CodeCommit
- Certificati server: certificati SSL/TLS che è possibile utilizzare per l'autenticazione con alcuni servizi. AWS Ti consigliamo di utilizzare AWS Certificate Manager (ACM) per fornire, gestire e distribuire i certificati del server. Utilizza IAM solo quando è necessario il supporto alle connessioni HTTPS in una regione che non è supportata da ACM. Per informazioni sulle regioni che supportano ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

È possibile scegliere le credenziali che meglio si adattano all'utente IAM. Quando utilizzi la AWS Management Console per creare un utente IAM, devi scegliere di includere almeno una password o delle chiavi di accesso alla console. Per impostazione predefinita, un nuovo utente IAM creato

utilizzando l' AWS API AWS CLI or non dispone di credenziali di alcun tipo. Il tipo di credenziali dell'utente IAM da creare dipende dal caso d'uso.

Hai le seguenti opzioni per amministrare le password, le chiavi di accesso e i dispositivi con l'autenticazione a più fattori (MFA):

- Gestione di password per gli utenti IAM. Crea e modifica le password che consentono l'accesso alla AWS Management Console. Imposta una policy per la password, così da implementare un minimo di complessità per la password. Consenti agli utenti di cambiare le loro password.
- Gestione delle chiavi di accesso per gli utenti IAM. Crea e aggiorna le chiavi di accesso per l'accesso programmatico alle risorse nel tuo account.
- Abilita l'utente IAM all'autenticazione a più fattori (MFA). Come best practice, ti consigliamo di richiedere l'autenticazione a più fattori per tutti gli utenti IAM nel tuo account. Con l'MFA, gli utenti devono fornire due forme di identificazione. Innanzitutto, forniscono le credenziali che fanno parte dell'identità utente (una password o una chiave di accesso). Inoltre, forniscono un codice numerico temporaneo generato su un dispositivo hardware o da un'applicazione su uno smartphone o un tablet.
- Trovare password e chiavi di accesso non utilizzate. Chiunque disponga di una password o di chiavi di accesso per il tuo account o di un utente IAM nel tuo account ha accesso alle tue AWS risorse. La sicurezza delle best practice consiste nel rimuovere le password e le chiavi di accesso quando gli utenti non ne hanno più bisogno.
- Download di un report delle credenziali per l'account. È possibile generare e scaricare un report delle credenziali che riporta tutti gli utenti IAM presenti nell'account e lo stato delle loro diverse credenziali, tra cui password, chiavi di accesso e dispositivi MFA. Per le password e le chiavi di accesso, il report sulle credenziali mostra se la password o la chiave di accesso siano state utilizzate di recente.

Utenti e autorizzazioni IAM

Per impostazione predefinita, un nuovo utente IAM non ha le autorizzazioni per svolgere alcuna operazione. Non sono autorizzati a eseguire alcuna AWS operazione o ad accedere a nessuna AWS risorsa. Un vantaggio di avere singoli utenti IAM è quello di poter assegnare le autorizzazioni individualmente a ogni utente. Potresti assegnare autorizzazioni amministrative a pochi utenti, che quindi possono amministrare AWS le tue risorse e possono persino creare e gestire altri utenti IAM. Nella maggior parte dei casi, tuttavia, desideri limitare le autorizzazioni di un utente solo alle attività (AWS azioni o operazioni) e alle risorse necessarie per il lavoro.

Prendiamo a esempio un utente denominato Diego. Quando crei l'utente IAM `Diego`, crei una password per questo utente e colleghi le autorizzazioni all'utente, per permettergli di avviare una determinata istanza Amazon EC2 e leggere le informazioni (`GET`) da una tabella in un database Amazon RDS. Per le procedure su come creare gli utenti e concedere loro le credenziali iniziali e le autorizzazioni, consulta Creare un utente IAM nel tuo Account AWS. Per le procedure su come modificare le autorizzazioni agli utenti esistenti, consulta Modifica delle autorizzazioni per un utente IAM. Per le procedure su come cambiare la password dell'utente o le chiavi di accesso, consulta la pagina Gestione delle password degli utenti in AWS e Gestione delle chiavi di accesso per gli utenti IAM.

Puoi anche aggiungere un limite delle autorizzazioni agli utenti IAM. Un limite di autorizzazioni è una funzionalità avanzata che consente di utilizzare policy AWS gestite per limitare le autorizzazioni massime che una policy basata sull'identità può concedere a un utente o un ruolo IAM. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Policy e autorizzazioni in IAM.

Utenti IAM e account

Ogni utente IAM è associato a un solo Account AWS. Poiché gli utenti IAM sono definiti all'interno del tuo account Account AWS, non è necessario che abbiano un metodo di pagamento registrato. AWS Qualsiasi AWS attività svolta dagli utenti IAM nel tuo account viene fatturata sul tuo account.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Utenti IAM come account di servizio

Un utente IAM è una risorsa in IAM con credenziali e autorizzazioni associate. Un utente IAM può rappresentare una persona o un'applicazione che utilizza le proprie credenziali per effettuare richieste AWS . In genere questo si chiama account di servizio. Se nella tua applicazione scegli di utilizzare le credenziali a lungo termine di un utente IAM, non integrare le chiavi di accesso direttamente nel codice dell'applicazione. Gli AWS SDK e gli SDK AWS Command Line Interface consentono di inserire le chiavi di accesso in posizioni note in modo da non doverle conservare nel codice. Per ulteriori informazioni, consulta Gestione corretta delle chiavi di accesso dell'utente IAM nella Riferimenti generali di AWS. Oppure, come best practice, puoi utilizzare le credenziali di sicurezza temporanee (ruoli IAM) al posto delle chiavi di accesso a lungo termine.

Creare un utente IAM nel tuo Account AWS

> **Important**
>
> Le migliori pratiche IAM consigliano di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee anziché utilizzare utenti IAM con credenziali a lungo termine.

> **Note**
>
> Se hai trovato questa pagina perché stai cercando informazioni su Product Advertising API per vendere prodotti Amazon sul tuo sito Web, consulta la documentazione relativa a Product Advertising API 5.0.
> Se sei arrivato a questa pagina dalla console IAM, è possibile che il tuo account non includa gli utenti IAM, anche se hai eseguito l'accesso. È possibile che tu abbia effettuato l'accesso come Utente root dell'account AWS, utilizzando un ruolo o che abbia utilizzato le credenziali provvisorie per accedere. Per ulteriori informazioni su queste identità IAM, consulta Identità IAM (utenti, gruppi di utenti e ruoli).

Il processo con cui si crea un utente e gli si consente di eseguire attività lavorative consiste nelle fasi seguenti:

1. Crea l'utente negli AWS Management Console strumenti per Windows PowerShell o utilizzando un'operazione AWS API. AWS CLI Se crei l'utente in AWS Management Console, i passaggi da 1 a 4 vengono gestiti automaticamente, in base alle tue scelte. Se crei gli utenti in modo programmatico, allora devi eseguire ognuna di queste fasi individualmente.
2. Creazione delle credenziali per l'utente, a seconda del tipo di accesso che l'utente richiede:
   - Abilita l'accesso alla console: facoltativo: se l'utente deve accedere a AWS Management Console, crea una password per l'utente. La disabilitazione dell'accesso alla console per un utente gli impedisce di accedere alla AWS Management Console tramite il nome utente e la password. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto.

> **Tip**
>
> Crea solo le credenziali di cui l'utente necessita. Ad esempio, per un utente che richiede l'accesso solo tramite AWS Management Console, non creare chiavi di accesso.

3. Concedi all'utente le autorizzazioni per l'esecuzione delle attività richieste aggiungendolo a uno o più gruppi. Puoi concedere autorizzazioni anche collegando le policy di autorizzazione direttamente all'utente. Tuttavia, consigliamo invece di inserire gli utenti in gruppi e gestire le autorizzazioni tramite le policy collegate a tali gruppi. È inoltre possibile utilizzare un limite delle autorizzazioni per limitare le autorizzazioni che un utente può avere, anche se questa pratica non è comune.
4. (Facoltativo) Aggiungere metadati all'utente collegando tag. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
5. Fornisci all'utente le necessarie informazioni di accesso. Queste informazioni includono la password e l'URL della console per la pagina di accesso all'account in cui l'utente immette tali credenziali. Per ulteriori informazioni, consulta In che modo gli utenti IAM accedono a AWS.
6. (Facoltativo) Configura multi-factor authentication (MFA) per l'utente. La MFA richiede all'utente di fornire un one-time-use codice ogni volta che accede a. AWS Management Console
7. (Facoltativo) Fornisci agli utenti le autorizzazioni per gestire le proprie credenziali di sicurezza. (Per impostazione predefinita, gli utenti non dispongono delle autorizzazioni per gestire le proprie credenziali). Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password.

Per informazioni sulle autorizzazioni di cui hai bisogno per creare un utente, consulta la pagina Autorizzazioni necessarie per accedere alle risorse IAM.

Argomenti

- Creazione di utenti IAM (console)
- Creazione di utenti IAM (AWS CLI)
- Creazione di utenti IAM (AWS API)

Creazione di utenti IAM (console)

Puoi utilizzare il AWS Management Console per creare utenti IAM.

Creazione di un utente IAM (console)

1. Segui la procedura di accesso appropriata al tuo tipo di utente, come descritto nell'argomento Come accedere ad AWS nella Guida per l'utente di AWS Sign-In.
2. Nella home page della console, seleziona il servizio IAM.
3. Nel pannello di navigazione seleziona Users (Utenti), quindi seleziona Add users (Aggiungi utenti).
4. Nella pagina Specify user details (Specifica dettagli utente), in User details (Dettagli utente), in User name (Nome utente), immetti il nome del nuovo utente. Questo è il nome di accesso per AWS.

   > **Note**
   >
   > Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote. I nomi utente possono essere una combinazione di un massimo di 64 lettere, cifre e i seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), trattino basso (_) e trattino (-). I nomi devono essere univoci nell'account. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare due utenti chiamati TESTUSER e testuser. Quando un nome utente viene utilizzato in una policy o come parte di un ARN, il nome fa distinzione tra maiuscole e minuscole. Quando un nome utente viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome utente non fa distinzione tra maiuscole e minuscole.

5. Seleziona Fornisci l'accesso utente al AWS Management Console (facoltativo) Questo produce le credenziali di AWS Management Console accesso per il nuovo utente.

   Ti verrà chiesto se stai fornendo l'accesso alla console a una persona. Ti consigliamo di creare utenti nel Centro identità IAM anziché in IAM.

   - Per passare alla creazione dell'utente nel Centro identità IAM, seleziona Specify a user in Identity Center (Specifica un utente nel Centro identità).

     Se non hai abilitato il Centro identità IAM, selezionando questa opzione verrai indirizzato alla pagina del servizio nella console in cui puoi abilitare il servizio. Per i dettagli su questa procedura, consulta la Guida introduttiva alle attività comuni in IAM Identity Center nella Guida per l'utente AWS IAM Identity Center

Se hai abilitato il Centro identità IAM, selezionando questa opzione verrai indirizzato alla pagina Specify user details (Specifica dettagli utente) nel Centro identità IAM. Per i dettagli su questa procedura, consulta Aggiungere utenti nella Guida AWS IAM Identity Center per l'utente

- Se non puoi usare il Centro identità IAM, seleziona I want to create an IAM user (Desidero creare un utente IAM) e continua a seguire questa procedura.

a. Per Console password (Password console), scegli una delle opzioni seguenti:

   - Autogenerated password (Password autogenerata): l'utente ottiene una password casuale che soddisfa la policy per le password dell'account. È possibile visualizzare o scaricare le password quando si arriva alla pagina Retrieve password (Recupera password).
   - Custom password (Password personalizzata): a ogni utente viene assegnata la password da digitare nella casella.

b. (Facoltativo) Per impostazione predefinita, l'opzione Users must create a new password at next sign-in (recommended) (Gli utenti devono creare una nuova password al prossimo accesso [consigliato]) è selezionata per essere certi che l'utente sia costretto a modificare la sua password al primo accesso.

> **Note**
>
> Se un amministratore ha attivato l'impostazione di policy per le password dell'account Allow users to change their own password (Consenti a tutti gli utenti di cambiare la loro password), questa casella di controllo non esegue alcuna operazione. In caso contrario, viene allegata automaticamente una policy AWS gestita denominata `IAMUserChangePassword` ai nuovi utenti. La policy concede agli utenti l'autorizzazione a modificare le proprie password.

6. Seleziona Avanti.
7. Nella pagina Set permissions (Imposta autorizzazioni), specifica il modo in cui desideri assegnare le autorizzazioni a questo utente. Seleziona una delle seguenti tre opzioni:

   - Add user to group (Aggiungi utente al gruppo): seleziona questa opzione se desideri assegnare l'utente a uno o più gruppi che hanno già le policy di autorizzazione. IAM mostra un elenco dei gruppi nell'account, insieme alle loro policy collegate. Puoi selezionare uno o più

gruppi esistenti oppure selezionare Create group (Crea gruppo) per creare un nuovo gruppo. Per ulteriori informazioni, consulta Modifica delle autorizzazioni per un utente IAM.

- Copy permissions (Copia autorizzazioni): seleziona questa opzione per copiare tutte le appartenenze ai gruppi, le policy gestite collegate, le policy in linea integrate e qualsiasi limite delle autorizzazioni esistente da un utente esistente al nuovo utente. IAM mostra l'elenco degli utenti nel tuo account. Seleziona quello le cui autorizzazioni corrispondono il più possibile alle esigenze del nuovo utente.
- Allega direttamente le politiche: seleziona questa opzione per visualizzare un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Seleziona le policy che desideri collegare all'utente oppure scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy. Per ulteriori informazioni, consulta la fase 4 nella procedura Creazione di policy IAM. Una volta creata la policy, chiudi la scheda e torna alla scheda originale per aggiungere la policy all'utente.

> **Tip**
>
> Quando possibile, collega le policy per un gruppo, quindi rendi gli utenti membri dei gruppi appropriati.

8. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

   Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni). IAM visualizza un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Seleziona la policy da utilizzare per il limite delle autorizzazioni o scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy. Per ulteriori informazioni, consulta la fase 4 nella procedura Creazione di policy IAM. Una volta creata la policy, chiudi la scheda e torna alla scheda originale per selezionare la policy da utilizzare per il limite delle autorizzazioni.
9. Seleziona Avanti.
10. (Facoltativo) Nella pagina Review and create (Rivedi e crea), in Tags (Tag), seleziona Add new tag (Aggiungi nuovo tag) per aggiungere i metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
11. Rivedi tutte le opzioni scelte fino a questo momento. Una volta pronto per continuare, seleziona Create user (Crea utente).

12. Nella pagina Retrieve password (Recupera password), ottieni la password assegnata all'utente:
    - Seleziona Show (Mostra) accanto alla password per visualizzare la password dell'utente in modo da poterla registrare manualmente.
    - Seleziona Download .csv (Scarica .csv) per scaricare le credenziali di accesso dell'utente come file .csv da salvare in una posizione sicura.
13. Seleziona Email sign-in instructions (Istruzioni di accesso via e-mail). Il client di posta elettronica locale si apre con una bozza che è possibile personalizzare e inviare all'utente. Il modello dell'e-mail include i seguenti dettagli per ciascun utente:
    - Nome utente
    - URL della pagina per l'accesso all'account. Utilizza il seguente esempio, sostituendo il numero ID dell'account corretto o l'alias dell'account:

      ```
      https://AWS-account-ID or alias.signin.aws.amazon.com/console
      ```

    > **Important**
    >
    > La password dell'utente non è inclusa nel messaggio generato. È necessario fornirla all'utente rispettando le linee guida sulla sicurezza dell'organizzazione.

14. Se l'utente richiede anche chiavi di accesso per l'accesso programmatico, consulta. Gestione delle chiavi di accesso per gli utenti IAM

Creazione di utenti IAM (AWS CLI)

Puoi utilizzare il AWS CLI per creare un utente IAM.

Come creare un utente IAM (AWS CLI)

1. Creare un utente.
    - aws iam create-user
2. (Facoltativo) Concedere all'utente l'accesso alla AWS Management Console. Ciò richiede una password. È anche necessario fornite all'utente l'URL della pagina di accesso all'account.
    - era iam create-login-profile

3. (Facoltativo) Concedere all'utente l'accesso a livello di programmazione. Ciò richiede le chiavi di accesso.

   - era io create-access-key
   - Strumenti per Windows PowerShell: New-iam AccessKey
   - API IAM: CreateAccessKey

   > **Important**
   >
   > Questa è la tua unica opportunità per visualizzare o scaricare le chiavi di accesso segrete e devi fornire queste informazioni agli utenti prima che possano utilizzare l' AWS API. Salva i nuovi ID chiave di accesso e Secret Access Key dell'utente in un luogo sicuro. Successivamente a questa fase non sarà più possibile accedere alle chiavi segrete.

4. Aggiungere l'utente a uno o più gruppi. I gruppi specificati devono avere le policy collegate che concedono le autorizzazioni appropriate per l'utente.

   - era io add-user-to-group

5. (Facoltativo) Collegare una policy all'utente che definisca le autorizzazioni dell'utente. Nota: consigliamo di gestire le autorizzazioni dell'utente aggiungendo l'utente a un gruppo e collegando una policy al gruppo invece di collegarla direttamente all'utente.

   - era io attach-user-policy

6. (Facoltativo) Aggiungere attributi personalizzati all'utente collegando tag. Per ulteriori informazioni, consulta Gestione dei tag sugli utenti IAM (AWS CLI o AWS API).
7. (Facoltativo) Concedere le autorizzazioni utente per gestire le credenziali di sicurezza. Per ulteriori informazioni, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Creazione di utenti IAM (AWS API)

Puoi utilizzare l' AWS API per creare un utente IAM.

Per creare un utente IAM dalla (AWS API)

1. Creare un utente.

- CreateUser

2. (Facoltativo) Concedere all'utente l'accesso alla AWS Management Console. Ciò richiede una password. È anche necessario fornite all'utente l'URL della pagina di accesso all'account.

   - CreateLoginProfile

3. (Facoltativo) Concedere all'utente l'accesso a livello di programmazione. Ciò richiede le chiavi di accesso.

   - CreateAccessKey

   > **Important**
   >
   > Questa è la tua unica opportunità per visualizzare o scaricare le chiavi di accesso segrete e devi fornire queste informazioni agli utenti prima che possano utilizzare l' AWS API. Salva i nuovi ID chiave di accesso e Secret Access Key dell'utente in un luogo sicuro. Successivamente a questa fase non sarà più possibile accedere alle chiavi segrete.

4. Aggiungere l'utente a uno o più gruppi. I gruppi specificati devono avere le policy collegate che concedono le autorizzazioni appropriate per l'utente.

   - AddUserToGroup

5. (Facoltativo) Collegare una policy all'utente che definisca le autorizzazioni dell'utente. Nota: consigliamo di gestire le autorizzazioni dell'utente aggiungendo l'utente a un gruppo e collegando una policy al gruppo invece di collegarla direttamente all'utente.

   - AttachUserPolicy

6. (Facoltativo) Aggiungere attributi personalizzati all'utente collegando tag. Per ulteriori informazioni, consulta Gestione dei tag sugli utenti IAM (AWS CLI o AWS API).

7. (Facoltativo) Concedere le autorizzazioni utente per gestire le credenziali di sicurezza. Per ulteriori informazioni, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Controllo dell'accesso utente IAM alla AWS Management Console

Gli utenti IAM con autorizzazione che accedono a te Account AWS tramite il AWS Management Console possono accedere alle tue risorse. AWS L'elenco seguente mostra i modi in cui puoi concedere agli utenti IAM l'accesso alle tue Account AWS risorse tramite AWS Management Console. Mostra anche come gli utenti IAM possono accedere ad altre funzionalità AWS dell'account tramite il AWS sito web.

> **Note**
>
> L'uso di IAM non comporta alcun costo.

Il AWS Management Console

È possibile creare una password per ciascun utente IAM che deve accedere alla AWS Management Console. Gli utenti accedono alla console tramite la pagina di Account AWS accesso abilitata a IAM. Per informazioni su come visualizzare la pagina di accesso, consulta Come accedere ad AWS nella Guida per l'utente di Accedi ad AWS . Per informazioni sulla creazione di password , consulta Gestione delle password degli utenti in AWS.

Puoi impedire a un utente IAM di accedere a AWS Management Console rimuovendo la sua password. Ciò impedisce loro di accedere AWS Management Console utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l' AWS CLI accesso tramite Tools for Windows PowerShell, AWS API o AWS Console Mobile Application.

AWS Le tue risorse, come istanze Amazon EC2, bucket Amazon S3 e così via

Anche se gli utenti IAM dispongono di password, hanno ancora bisogno dell'autorizzazione per accedere alle risorse AWS . Quando crei un utente IAM, questo non dispone di autorizzazioni per impostazione predefinita. Per assegnare agli utenti IAM le autorizzazioni necessarie, puoi associare loro delle policy. Se disponi di molti utenti IAM che eseguiranno le stesse attività con le stesse risorse, puoi assegnare tali utenti IAM a un gruppo. Quindi assegna le autorizzazioni a tale gruppo. Per informazioni sulla creazione di utenti e gruppi IAM, vedere Identità IAM (utenti, gruppi di utenti e ruoli). Per ulteriori informazioni sull'utilizzo di policy per impostare le autorizzazioni, vedere Gestione degli accessi AWS alle risorse.

AWS Forum di discussione

Chiunque può leggere i post sui forum di discussione AWS. Gli utenti che desiderano pubblicare domande o commenti sul Forum di AWS discussione possono farlo utilizzando il proprio nome utente. La prima volta che un utente pubblica un post sul Forum di AWS discussione, gli viene richiesto di inserire un soprannome e un indirizzo e-mail. Solo quell'utente può usare quel soprannome nei forum di AWS discussione.

Le tue informazioni Account AWS di fatturazione e utilizzo

Puoi consentire agli utenti di accedere alle tue informazioni di Account AWS fatturazione e utilizzo. Per ulteriori informazioni, consulta Controllo dell'accesso alle informazioni di fatturazione nella Guida per l'utente di AWS Billing .

Le informazioni del tuo Account AWS profilo

Gli utenti non possono accedere alle informazioni Account AWS del tuo profilo.

Le tue credenziali Account AWS di sicurezza

Gli utenti non possono accedere alle tue credenziali Account AWS di sicurezza.

> **Note**
>
> Le policy IAM controllano l'accesso indipendentemente dall'interfaccia. Ad esempio, è possibile fornire a un utente la password per accedere alla AWS Management Console. Le policy per tale utente (o per qualsiasi gruppo cui l'utente appartiene) determina ciò che l'utente può fare nella AWS Management Console. In alternativa, puoi fornire all'utente le chiavi di AWS accesso a cui effettuare AWS chiamate API. In questo caso, le policy controllano le operazioni che l'utente può richiamare tramite una libreria o un client che utilizza le chiavi di accesso per l'autenticazione.

In che modo gli utenti IAM accedono a AWS

Per accedere AWS Management Console come utente IAM, devi fornire l'ID o l'alias dell'account oltre al nome utente e alla password. Quando l'amministratore ha creato l'utente IAM nella console, dovrebbero aver inviato le credenziali di accesso, ovvero nome utente e URL della pagina di accesso dell'account che include l'ID account o l'alias dell'account.

```
https://My_AWS_Account_ID.signin.aws.amazon.com/console/
```

> **Suggerimento**
>
> Per creare un segnalibro per la pagina di accesso al tuo account nel tuo browser Web, devi digitare manualmente l'URL della pagina di accesso per il tuo account nella voce segnalibro. Non utilizzare la funzione di segnalibro del tuo browser Web perché gli indirizzamenti possono oscurare l'URL della pagina di accesso.

Puoi effettuare l'accesso anche al seguente endpoint generale di accesso e digitare manualmente l'ID account o l'alias dell'account:

```
https://console.aws.amazon.com/
```

Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. La volta successiva che l'utente accede a qualsiasi pagina di AWS Management Console, la console utilizza il cookie per reindirizzare l'utente alla pagina di accesso dell'account.

Hai accesso solo alle AWS risorse specificate dall'amministratore nella policy allegata alla tua identità utente IAM. Per lavorare nella console, devi disporre delle autorizzazioni necessarie per eseguire le azioni eseguite dalla console, come elencare e creare AWS risorse. Per ulteriori informazioni, consulta Gestione degli accessi AWS alle risorse e Esempi di policy basate su identità IAM.

> **Note**
>
> Se la tua organizzazione dispone di un suo sistema di identità, puoi pensare di creare un'opzione Single Sign-On (SSO). L'SSO consente agli utenti di accedere al AWS Management Console tuo account senza richiedere loro di avere un'identità utente IAM. L'SSO elimina inoltre la necessità per gli utenti di accedere al sito dell'organizzazione e di farlo AWS separatamente. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

Registrazione dei dettagli di accesso in CloudTrail

Se abiliti CloudTrail la registrazione degli eventi di accesso nei tuoi registri, devi sapere come CloudTrail scegli dove registrare gli eventi.

- Se gli utenti accedono direttamente a una console, vengono reindirizzati a un endpoint di accesso globale o regionale, a seconda se la console del servizio selezionato supporti o meno le regioni. Ad esempio, la homepage della console principale supporta le regioni, perciò se effettui l'accesso al seguente URL:

  ```
  https://alias.signin.aws.amazon.com/console
  ```

  vieni reindirizzato a un dispositivo di accesso regionale, ad esempio`https://us-east-2.signin.aws.amazon.com`, con una voce di CloudTrail registro regionale nel registro dell'area dell'utente:

  D'altra parte, la console Amazon S3 non supporta le regioni, perciò se si effettua l'accesso al seguente URL:

  ```
  https://alias.signin.aws.amazon.com/console/s3
  ```

  AWS reindirizza l'utente all'endpoint di accesso globale all'indirizzo`https://signin.aws.amazon.com`, generando una voce di registro globale. CloudTrail
- È possibile richiedere manualmente un determinato endpoint di accesso regionale tramite l'accesso alla homepage della console principale con attivazione regionale, utilizzando una sintassi di URL simile a quella riportata di seguito:

  ```
  https://alias.signin.aws.amazon.com/console?region=ap-southeast-1
  ```

  AWS reindirizza l'utente all'endpoint di accesso `ap-southeast-1` regionale e genera un evento di registro regionale. CloudTrail

Per ulteriori informazioni su CloudTrail e IAM, consulta Registrazione degli eventi IAM con. CloudTrail

Se gli utenti richiedono un accesso programmatico per funzionare con il tuo account, puoi creare una coppia di chiavi di accesso (un ID chiave di accesso e una chiave di accesso segreta) per ciascun utente, come descritto in . Tuttavia, ci sono alternative più sicure da considerare prima di creare le chiavi di accesso per gli utenti. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella Riferimenti generali di AWS.

Utilizzo di dispositivi MFA con la pagina di accesso IAM

Gli utenti configurati con dispositivi di autenticazione a più fattori (MFA) devono utilizzare i propri dispositivi MFA per accedere alla AWS Management Console. Dopo che l'utente ha inserito le proprie credenziali di accesso, AWS controlla l'account dell'utente per vedere se l'MFA è richiesta per quell'utente. Gli argomenti seguenti forniscono informazioni su come gli utenti completano l'accesso quando è necessaria l'autenticazione MFA.

Argomenti

Accesso con più dispositivi MFA abilitati

Se un utente accede AWS Management Console come utente Account AWS root o utente IAM con più dispositivi MFA abilitati per quell'account, deve utilizzare un solo dispositivo MFA per accedere. Dopo l'autenticazione con la password dell'utente, l'utente seleziona il tipo di dispositivo MFA che desidera utilizzare per completare l'autenticazione. Quindi all'utente viene richiesto di autenticarsi con il tipo di dispositivo selezionato.

Accesso con una chiave di sicurezza FIDO

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. L'utente deve toccare la chiave di sicurezza FIDO.

> **Note**
>
> Gli utenti di Chrome non devono scegliere alcuna delle opzioni disponibili nel pop-up di Google Chrome che chiede di verificare la tua identità con amazon.com. Limitati a toccare la chiave di sicurezza.

A differenza di altri dispositivi MFA, le chiavi di sicurezza FIDO sono sempre aggiornate. Se una chiave di sicurezza FIDO viene smarrita o danneggiata, gli amministratori possono disattivarla. Per ulteriori informazioni, consulta Disattivazione dei dispositivi MFA (console).

Per informazioni sui browser che supportano WebAuthn e sui dispositivi compatibili con FIDO che supportano, vedere. AWS Configurazioni supportate per l'utilizzo di chiavi di accesso e chiavi di sicurezza

Accesso con un dispositivo MFA virtuale

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. Nella casella MFA code (Codice MFA), l'utente deve immettere il codice numerico fornito dall'applicazione MFA.

Se il codice MFA è corretto, l'utente può accedere alla AWS Management Console. Se il codice non è corretto, l'utente può riprovare con un altro codice.

Un dispositivo MFA virtuale può andare fuori sincrono. Se un utente non riesce ad accedere AWS Management Console dopo diversi tentativi, all'utente viene richiesto di sincronizzare il dispositivo MFA virtuale. L'utente può seguire le istruzioni mostrate sullo schermo per sincronizzare il dispositivo MFA virtuale. Per informazioni su come sincronizzare un dispositivo per conto di un utente del tuo paese, consulta. Account AWSRisincronizzazione dei dispositivi MFA virtuali e hardware

Accesso con un token TOTP hardware

Se l'autenticazione MFA è obbligatorio per l'utente, viene visualizzata una seconda pagina di accesso. Nella casella MFA code (Codice MFA), l'utente deve immettere il codice numerico fornito dal token TOTP hardware.

Se il codice MFA è corretto, l'utente può accedere alla AWS Management Console. Se il codice non è corretto, l'utente può riprovare con un altro codice.

Un token TOTP hardware può non essere sempre sincronizzato. Se un utente non riesce ad accedere AWS Management Console dopo diversi tentativi, all'utente viene richiesto di sincronizzare il dispositivo token MFA. L'utente può seguire le istruzioni visualizzate per sincronizzare il dispositivo token MFA. Per informazioni su come sincronizzare un dispositivo per conto di un utente del tuo, consulta. Account AWSRisincronizzazione dei dispositivi MFA virtuali e hardware

Gestione degli utenti IAM

> **Note**
>
> Come procedura consigliata, si consiglia di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se segui le best practice, non gestisci utenti e gruppi IAM. Gli utenti e i gruppi

sono invece gestiti all'esterno AWS e possono accedere alle AWS risorse come identità federata. Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web, il AWS Directory Service, la directory Identity Center o qualsiasi utente che accede AWS ai servizi utilizzando le credenziali fornite tramite un'origine di identità. Le identità federate utilizzano i gruppi definiti dal rispettivo gestore di identità. Se lo utilizzi AWS IAM Identity Center, consulta Gestisci le identità in IAM Identity Center nella Guida per l'AWS IAM Identity Center utente per informazioni sulla creazione di utenti e gruppi in IAM Identity Center.

Amazon Web Services offre vari strumenti per gestire gli utenti IAM nel proprio Account AWS. Puoi elencare gli utenti IAM nel tuo account o in un gruppo di utenti oppure elencare tutti i gruppi di utenti di cui un utente è membro. È possibile rinominare o modificare il percorso di un utente IAM. Se desideri utilizzare le identità federate invece degli utenti IAM, puoi eliminare un utente IAM dall'account AWS o disattivarlo.

Per ulteriori informazioni sull'aggiunta, la modifica o la rimozione di policy gestite per un utente IAM, consulta Modifica delle autorizzazioni per un utente IAM. Per informazioni sulla gestione di policy in linea per utenti IAM, consulta Aggiunta e rimozione di autorizzazioni per identità IAM, Modifica delle policy IAM e Eliminazione di policy IAM. Come best practice, utilizza le policy gestite anziché le policy in linea. Le policy gestite da AWS concedono autorizzazioni per numerosi casi d'uso comuni. Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per l'uso da parte di tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo Policy gestite dal cliente specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta AWS politiche gestite. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, consulta. AWS politiche gestite per le funzioni lavorative

Per ulteriori informazioni sulla convalida delle policy IAM, consulta Convalida delle policy IAM.

Tip

IAM Access Analyzer analizza i servizi e le azioni utilizzati dai tuoi ruoli IAM e quindi genera una policy dettagliata che puoi utilizzare. Dopo aver testato ogni policy generata, puoi distribuirla nell'ambiente di produzione. In questo modo si garantisce di concedere solo le autorizzazioni necessarie ai carichi di lavoro. Per ulteriori informazioni sulla generazione delle policy, consulta IAM Access Analyzer policy generation.

Per informazioni sulla gestione delle password utente IAM, consulta Gestione delle password per gli utenti IAM,

Argomenti

- Visualizzazione dell'accesso utente
- Elenco di utenti IAM
- Ridenominazione di un utente IAM
- Eliminazione di un utente IAM
- Disattivazione di un utente IAM

Visualizzazione dell'accesso utente

Prima di eliminare un utente, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Elenco di utenti IAM

Puoi elencare gli utenti IAM del tuo Account AWS o di uno specifico gruppo di utenti IAM ed elencare tutti i gruppi di utenti a cui appartiene un utente. Per informazioni sulle autorizzazioni necessarie per elencare gli utenti, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Per elencare gli utenti nell'account.

- AWS Management ConsoleNel riquadro di navigazione, seleziona Users (Utenti). La console mostra gli utenti del tuo Account AWS.
- AWS CLI: aws iam list-users
- AWS API: ListUsers

Come elencare gli utenti in un determinato gruppo di utenti

- AWS Management Console: nel pannello di navigazione, scegli Gruppi di utenti, seleziona il nome del gruppo di utenti quindi scegli la scheda Utenti.
- AWS CLI: aws iam get-group
- AWS API: GetGroup

Come elencare tutti i gruppi di utenti in cui si trova un utente

- AWS Management Console: nel riquadro di navigazione, scegliere Users (Utenti), il nome utente e selezionare la scheda Gruppi.
- AWS CLI: Aws Iam list-groups-for-user
- AWS API: ListGroupsForUser

Ridenominazione di un utente IAM

Per modificare il nome o il percorso di un utente, è necessario utilizzare Tools for Windows PowerShell o AWS l'API. AWS CLI Non è disponibile alcuna opzione nella console per rinominare un utente. Per informazioni sulle autorizzazioni necessarie per ridenominare un utente, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Quando si modifica il nome o il percorso di un utente, si verificano i seguenti eventi:

- Qualsiasi policy collegata all'utente viene mantenuta per l'utente con il nuovo nome.
- L'utente rimane negli stessi gruppi di utenti con il nuovo nome.
- L'ID univoco dell'utente rimane invariato. Per ulteriori informazioni sugli ID univoci, consulta Identificatori univoci.
- Qualsiasi policy relativa alle risorse o ai ruoli che fa riferimento all'utente come principale (l'utente a cui viene consentito l'accesso) viene automaticamente aggiornata per l'utilizzo del nuovo nome o percorso. Ad esempio, qualsiasi policy basata su code in Amazon SQS o basata sulle risorse in Amazon S3 viene aggiornata automaticamente per utilizzare il nuovo nome e percorso.

IAM non aggiorna automaticamente le policy che fanno riferimento all'utente come una risorsa per l'utilizzo del nuovo nome o percorso, è necessario un aggiornamento manuale. Ad esempio, si immagini che l'utente `Richard` disponga di una policy collegata che permette di gestire le credenziali di sicurezza dell'utente. Se un amministratore rinomina `Richard` in `Rich`, l'amministratore deve anche aggiornare questa policy per modificar la risorsa da:

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Richard
```

a:

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Rich
```

Ciò è valido anche se un amministratore cambia il percorso, l'amministratore deve aggiornare la policy in base al nuovo percorso per l'utente.

Per rinominare un utente

- AWS CLI: aws iam update-user
- AWS API: UpdateUser

Eliminazione di un utente IAM

Potresti eliminare un utente IAM dalla tua azienda Account AWS se quell'utente lascia la tua azienda. Se l'utente è temporaneamente assente, è possibile disattivare l'accesso dell'utente invece di eliminarlo dall'account come descritto nella sezione Disattivazione di un utente IAM.

Argomenti

- Eliminazione di un utente IAM (console)
- Eliminazione di un utente IAM (AWS CLI)

Eliminazione di un utente IAM (console)

Quando utilizzi il AWS Management Console per eliminare un utente IAM, IAM elimina automaticamente le seguenti informazioni per te:

- L'utente
- Qualsiasi appartenenza al gruppo, ovvero, l'utente viene rimosso da qualsiasi gruppo IAM di cui l'utente era membro
- Qualsiasi password associata all'utente
- Qualsiasi chiave di accesso di proprietà dell'utente
- Tutte le policy in linea integrate nell'utente (le policy applicate a un utente tramite le autorizzazioni del gruppo non sono interessate)

> **Note**
>
> IAM rimuove tutte le policy gestite collegate all'utente quando si elimina l'utente, ma non elimina le policy gestite.

- Qualsiasi dispositivo MFA associato

Per eliminare un gruppo IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Utenti, quindi la casella di controllo accanto al nome utente da eliminare.
3. Nella parte superiore della pagina, scegli Delete (Elimina).
4. Nella casella di dialogo di conferma, inserisci il nome utente nel campo di inserimento testo per confermare l'eliminazione dell'utente. Scegli Delete (Elimina).

Eliminazione di un utente IAM (AWS CLI)

A differenza di AWS Management Console, quando elimini un utente con AWS CLI, devi eliminare gli elementi associati all'utente manualmente. Questa procedura illustra il processo.

Per eliminare un utente dall'account (AWS CLI)

1. Eliminare la password dell'utente, se l'utente ne ha una.

   `aws iam delete-login-profile`
2. Eliminare le chiavi di accesso dell'utente, se disponibili.

   `aws iam list-access-keys` (per elencare le chiavi di accesso dell'utente) e `aws iam delete-access-key`
3. Eliminare il certificato di firma dell'utente. Si noti che quando si eliminano delle credenziali di sicurezza queste vengono viene eliminate per sempre e non possono più essere recuperate.

   `aws iam list-signing-certificates` (per elencare i certificati di firma dell'utente) e `aws iam delete-signing-certificate`
4. Eliminare la chiave pubblica SSH dell'utente, se disponibile.

   `aws iam list-ssh-public-keys` (per elencare le chiavi pubbliche SSH dell'utente) e `aws iam delete-ssh-public-key`
5. Eliminare le credenziali Git dell'utente.

   `aws iam list-service-specific-credentials` (per elencare le credenziali git dell'utente) e `aws iam delete-service-specific-credential`
6. Disattivare il dispositivo Multi-Factor Authentication (MFA), se uno è disponibile.

`aws iam list-mfa-devices` (per elencare i dispositivi MFA dell'utente), `aws iam deactivate-mfa-device` (per disattivare il dispositivo) e `aws iam delete-virtual-mfa-device` (per eliminare definitivamente un dispositivo MFA virtuale)

7. Eliminare le policy inline dell'utente.

   `aws iam list-user-policies` (per elencare le policy inline per l'utente) e `aws iam delete-user-policy` (per eliminare la policy)

8. Scollegare le policy gestite collegate all'utente.

   `aws iam list-attached-user-policies` (per elencare le policy gestite collegate all'utente) e `aws iam detach-user-policy` (per scollegare la policy)

9. Rimuovi l'utente da qualsiasi gruppo di utenti.

   `aws iam list-groups-for-user` (per elencare i gruppi di utenti a cui l'utente appartiene) e `aws iam remove-user-from-group`

10. Eliminare l'utente.

    `aws iam delete-user`

Disattivazione di un utente IAM

Potrebbe essere necessario disattivare un utente IAM mentre è temporaneamente lontano dall'azienda. Puoi lasciare invariate le loro credenziali utente IAM e bloccarne comunque AWS l'accesso.

Per disattivare un utente, crea e collega una policy per negare all'utente l'accesso a AWS. Puoi ripristinare l'accesso dell'utente in un secondo momento.

Di seguito sono riportati due esempi di policy di diniego che puoi collegare a un utente per negargli l'accesso.

La seguente policy non include un limite di tempo. È necessario rimuovere la policy per ripristinare l'accesso dell'utente.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
      {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*"
      }
   ]
}
```

La seguente policy include una condizione che avvia la policy il 24 dicembre 2024 alle 23:59 (UTC) e la termina il 28 febbraio 2025 alle 23:59 (UTC).

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
          "DateGreaterThan": {"aws:CurrentTime": "2024-12-24T23:59:59Z"},
          "DateLessThan": {"aws:CurrentTime": "2025-02-28T23:59:59Z"}
          }
       }
   ]
}
```

Modifica delle autorizzazioni per un utente IAM

Puoi modificare le autorizzazioni per un utente IAM del tuo paese Account AWS modificando l'appartenenza ai gruppi, copiando le autorizzazioni da un utente esistente, allegando le policy direttamente a un utente o impostando un limite di autorizzazioni. Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un utente. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Per informazioni sulle autorizzazioni necessarie per modificare le autorizzazioni per un utente, consulta la pagina Autorizzazioni necessarie per accedere alle risorse IAM.

Argomenti

- Visualizzazione dell'accesso utente
- Generazione di una policy basata sull'attività di accesso di un utente

- Aggiunta di autorizzazioni a un utente (console)
- Modifica delle autorizzazioni per un utente (console)
- Rimozione di una policy delle autorizzazioni da un utente (console)
- Rimozione di un limite delle autorizzazioni da un utente (console)
- Aggiungere e rimuovere le autorizzazioni (AWS CLI o AWS API) di un utente

Visualizzazione dell'accesso utente

Prima di modificare le autorizzazioni per un utente, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Generazione di una policy basata sull'attività di accesso di un utente

Talvolta, è possibile concedere autorizzazioni a un'entità IAM (utente o ruolo) oltre a quelle richieste. Per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy basate sull'attività di accesso.

Aggiunta di autorizzazioni a un utente (console)

Per aggiungere policy di autorizzazione a un utente, IAM offre tre diverse possibilità:

- Aggiungi utente al gruppo: rende l'utente membro di un gruppo. Le policy del gruppo vengono collegate all'utente.
- Copia le autorizzazioni dall'utente esistente: copia tutte le appartenenze ai gruppi, le policy gestite collegate, le policy in linea e tutti i limiti delle autorizzazioni esistenti dall'utente di origine.
- Collega direttamente le policy all'utente: collega una policy gestita direttamente all'utente. Per una gestione più semplice, si consiglia di assegnare le policy a un gruppo e rendere quindi gli utenti membri dei gruppi appropriati.

> **Important**
>
> Se l'utente ha un limite delle autorizzazioni, non è possibile aggiungere più autorizzazioni di quante ne consenta il limite delle autorizzazioni.

Aggiunta di autorizzazioni aggiungendo l'utente a un gruppo

L'aggiunta di un utente a un gruppo influisce immediatamente sull'utente.

Per aggiungere autorizzazioni aggiungendo l'utente a un gruppo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Riconsultare le appartenenze ai gruppi correnti per gli utenti nella colonna Groups (Gruppi) della console. Se necessario, aggiungere la colonna alla tabella degli utenti mediante la procedura seguente:

   1. Sopra la tabella a destra, selezionare il simbolo delle impostazioni ( ).
   2. Nella finestra di dialogo Manage Columns (Gestisci colonne) selezionare la colonna Groups (Gruppi). Facoltativamente, è anche possibile deselezionare la casella di controllo per le intestazioni di colonna che non si desidera visualizzare nella tabella utenti.
   3. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.

   La colonna Groups (Gruppi) indica i gruppi a cui appartiene l'utente. La colonna include i nomi dei gruppi per un massimo di due gruppi. Se l'utente è membro di tre o più gruppi, vengono visualizzati i primi due gruppi (ordinati in ordine alfabetico) e viene incluso il numero di appartenenze di gruppo aggiuntive. Ad esempio, se l'utente appartiene al Gruppo A, Gruppo B, Gruppo C e Gruppo D, il campo contiene il valore Group A, Group B + 2 more (Gruppo A, Gruppo B + 2 altri). Per visualizzare il numero totale di gruppi a cui appartiene l'utente, è possibile aggiungere la colonna Group count (Conteggio dei gruppi) alla tabella degli utenti.
4. Selezionare il nome dell'utente per cui modificare le autorizzazioni.
5. Selezionare la scheda Permissions (Autorizzazioni) e selezionare Add permissions (Aggiungi autorizzazioni). Scegli Add user to group (Aggiungi utente al gruppo).

6. Selezionare la casella di controllo per ciascun gruppo in cui si desidera includere l'utente. L'elenco mostra il nome di ciascun gruppo e le policy che l'utente riceve se diventa un membro di tale gruppo.
7. (Facoltativo) Oltre alla selezione da gruppi esistenti, è possibile selezionare Create group (Crea gruppo) per definire un nuovo gruppo:

   a. Nella nuova scheda, per User group name (Nome gruppo di utenti), digita un nome per il nuovo gruppo.

      > **Note**
      >
      > Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote. I nomi dei gruppi possono essere una combinazione di un massimo di 128 lettere, cifre e dei seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@) e trattino (-). I nomi devono essere univoci nell'account. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare due gruppi chiamati TESTGROUP e testgroup.

   b. Selezionare una o più caselle di controllo per le policy gestite da collegare al gruppo. È inoltre possibile creare una nuova policy gestita selezionando Create policy (Crea policy). In questo caso, tornare a questa scheda o finestra del browser quando la nuova policy è stata completata, selezionare Refresh (Aggiorna) e quindi selezionare la nuova policy da collegare al gruppo. Per ulteriori informazioni, consulta Creazione di policy IAM.

   c. Scegli Create user group (Crea gruppo di utenti).

   d. Tornare alla scheda originale e aggiornare l'elenco di gruppi. Quindi selezionare la casella di controllo del nuovo gruppo.

8. Scegli Next (Successivo) per visualizzare l'elenco dei membri del gruppo da aggiungere all'utente. Selezionare quindi Add permissions (Aggiungi autorizzazioni).

Aggiunta di autorizzazioni tramite copia da un altro utente

La copia delle autorizzazioni influisce immediatamente sull'utente.

Per aggiungere le autorizzazioni per un utente copiandole da un altro utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Selezionare Users (Utenti) nel riquadro di navigazione, selezionare il nome dell'utente per cui modificare le autorizzazioni e quindi selezionare la scheda Permissions (Autorizzazioni).
3. Selezionare Add permissions (Aggiungi autorizzazioni), quindi selezionare Copy permissions from existing user (Copia autorizzazioni dall'utente esistente). L'elenco mostra gli utenti disponibili con le relative appartenenze ai gruppi e le policy collegate. Se l'elenco completo dei gruppi o delle policy non rientra in un'unica riga, è possibile selezionare il collegamento per E altre ***n***. Questa operazione consente di aprire una nuova scheda del browser e visualizzare l'elenco completo delle policy nella scheda Permissions (Autorizzazioni) e gruppi nella scheda Groups (Gruppi).
4. Selezionare il pulsante di opzione accanto all'utente che dispone delle autorizzazioni che si desidera copiare.
5. Seleziona Next (Successivo) per visualizzare l'elenco delle modifiche che devono essere apportate all'utente. Selezionare quindi Add permissions (Aggiungi autorizzazioni).

Aggiunta di autorizzazioni collegando le policy direttamente all'utente

Il collegamento delle policy influisce immediatamente sull'utente.

Per aggiungere le autorizzazioni collegando direttamente le policy gestite all'utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Users (Utenti) nel riquadro di navigazione, selezionare il nome dell'utente per cui modificare le autorizzazioni e quindi selezionare la scheda Permissions (Autorizzazioni).
3. Scegli Add permissions (Aggiungi autorizzazioni), quindi seleziona Attach policies directly (Collega direttamente le policy).
4. Selezionare una o più caselle di controllo per le policy gestite da collegare all'utente. È inoltre possibile creare una nuova policy gestita selezionando Create policy (Crea policy). In questo caso, tornare a questa scheda o finestra del browser quando la nuova policy è stata completata. Selezionare Refresh (Aggiorna) e quindi selezionare la casella di controllo relativa alla nuova policy per collegarla all'utente. Per ulteriori informazioni, consulta Creazione di policy IAM.
5. Seleziona Next (Successivo) per visualizzare l'elenco delle policy che devono essere collegate all'utente. Selezionare quindi Add permissions (Aggiungi autorizzazioni).

Impostazione del limite delle autorizzazioni per un utente

L'impostazione di un limite delle autorizzazioni influisce immediatamente sull'utente.

Per impostare il limite delle autorizzazioni per un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Selezionare il nome dell'utente per cui modificare il limite delle autorizzazioni.
4. Scegli la scheda Autorizzazioni. Se necessario, apri la sezione Permissions boundary (Limite delle autorizzazioni), quindi seleziona Set permissions boundary (Imposta limite delle autorizzazioni).
5. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
6. Scegliere Set boundary (Imposta limite).

Modifica delle autorizzazioni per un utente (console)

IAM consente di modificare le autorizzazioni associate a un utente nei modi seguenti:

- Modifica una policy di autorizzazione: è possibile modificare la policy in linea di un utente, la policy in linea del gruppo dell'utente o la policy gestita collegato all'utente direttamente o da un gruppo. Se l'utente ha un limite delle autorizzazioni, non è possibile fornire più autorizzazioni di quante ne consenta la policy utilizzata come limite delle autorizzazioni dell'utente.
- Modifica del limite delle autorizzazioni: modifica la policy utilizzata come limite delle autorizzazioni per l'utente. In questo modo è possibile ampliare o limitare il numero massimo di autorizzazioni che un utente può avere.

Modifica di una policy di autorizzazione collegata a un utente

La modifica delle autorizzazioni influisce immediatamente sull'utente.

Per modificare le policy gestite collegate a un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.

3. Selezionare il nome dell'utente per cui modificare la policy di autorizzazione.
4. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions policies (Policy di autorizzazione).
5. Selezionare il nome della policy da modificare per visualizzare i relativi dettagli. Seleziona la scheda Policy usage (Utilizzo delle policy) per visualizzare le altre entità che potrebbero essere interessate dalla modifica della policy.
6. Selezionare la scheda Permissions (Autorizzazioni) e rivedere le autorizzazioni concesse dalla policy. Quindi selezionare Edit policy (Modifica policy).
7. Modifica la policy e risolvi eventuali suggerimenti di convalida della policy. Per ulteriori informazioni, consulta Modifica delle policy IAM.
8. Selezionare Review policy (Esamina policy), esaminare il riepilogo della policy, quindi selezionare Save changes (Salva le modifiche).

Modifica del limite delle autorizzazioni per un utente

La modifica del limite delle autorizzazioni influisce immediatamente sull'utente.

Per modificare la policy utilizzata per impostare il limite delle autorizzazioni per un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Selezionare il nome dell'utente per cui modificare il limite delle autorizzazioni.
4. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions boundary (Limite delle autorizzazioni) e selezionare Change boundary (Modifica limite).
5. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
6. Scegliere Set boundary (Imposta limite).

Rimozione di una policy delle autorizzazioni da un utente (console)

La rimozione di una policy influisce immediatamente sull'utente.

Revoca delle autorizzazioni per gli utenti IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, seleziona Utenti.
3. Selezionare il nome dell'utente per cui rimuovere il limite delle autorizzazioni.
4. Scegli la scheda Autorizzazioni.
5. Se desideri revocare le autorizzazioni rimuovendo una policy esistente, visualizza il tipo di policy per comprendere il modo in cui l'utente riceve la policy prima di selezionare X per rimuoverla:

   - Se la policy è applicata tramite l'appartenenza a un gruppo, seleziona X per rimuovere l'utente dal gruppo. Ricordare che potrebbero essere presenti più policy associate a un singolo gruppo. Se si rimuove un utente da un gruppo, l'utente perde l'accesso a tutte le policy che ricevute tramite l'appartenenza a tale gruppo.
   - Se la policy è una policy gestita collegata direttamente all'utente, scegliendo Remove (Rimuovi) questa sarà scollegata dall'utente. Ciò non influisce sulla policy stessa o su qualsiasi altra entità a cui la policy potrebbe essere collegata.
   - Se la policy è una policy in linea integrata, scegliendo X si rimuove la policy da IAM. Le policy inline collegate direttamente a un utente sono presenti solo in tale utente.

Rimozione di un limite delle autorizzazioni da un utente (console)

La rimozione del limite delle autorizzazioni influisce immediatamente sull'utente.

Per rimuovere il limite delle autorizzazioni da un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Selezionare il nome dell'utente per cui rimuovere il limite delle autorizzazioni.
4. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions boundary (Limite delle autorizzazioni) e quindi selezionare Remove boundary (Rimuovi limite).
5. Scegli Remove boundary (Rimuovi limite) per confermare la rimozione del limite delle autorizzazioni.

Aggiungere e rimuovere le autorizzazioni (AWS CLI o AWS API) di un utente

Per aggiungere o rimuovere le autorizzazioni a livello di codice, è necessario aggiungere o rimuovere i gruppi di appartenenza, collegare o distaccare le appartenenze ai gruppi, collegare o distaccare

le policy gestite oppure aggiungere o eliminare le policy inline. Per ulteriori informazioni, consulta i seguenti argomenti:

- Aggiunta e rimozione di utenti in un gruppo di utenti IAM
- Aggiunta e rimozione di autorizzazioni per identità IAM

Gestione delle password degli utenti in AWS

Puoi gestire le password per gli utenti IAM del tuo account. Gli utenti IAM necessitano di password per accedere a. AWS Management Console Gli utenti non hanno bisogno di password per accedere alle AWS risorse in modo programmatico utilizzando gli strumenti per Windows AWS CLI PowerShell, gli SDK o le AWS API. Per questi ambienti, hai la possibilità di assegnare chiavi di accesso agli utenti IAM. Tuttavia, ci sono altre alternative più sicure delle chiavi di accesso che ti consigliamo di prendere in considerazione per prime. Per ulteriori informazioni, consulta AWS credenziali di sicurezza.

Indice

Impostazione di una policy delle password dell'account per utenti IAM

Puoi impostare una politica di password personalizzata Account AWS per specificare i requisiti di complessità e i periodi di rotazione obbligatori per le password degli utenti IAM. Se non imposti una politica di password personalizzata, le password utente IAM devono soddisfare la politica di password predefinita AWS . Per ulteriori informazioni, consulta Opzioni di policy delle password personalizzata.

Argomenti

- Impostazione di una policy sulle password (AWS CLI)
- Impostazione di una politica in materia di password (AWS API)

Impostazione di una policy delle password

La policy sulle password IAM non si applica alla Utente root dell'account AWS password o alle chiavi di accesso utente IAM. Se una password scade, l'utente IAM non può accedere AWS Management Console ma può continuare a utilizzare le proprie chiavi di accesso.

Quando si crea o si modifica una policy sulle password, la maggior parte delle impostazioni sulla policy delle password vengono applicate la prossima volta che gli utenti modificano le password. Tuttavia, alcune delle impostazioni vengono applicate immediatamente. Ad esempio:

- Quando si impostano i requisiti minimi di lunghezza e del tipo di caratteri, le impostazioni vengono applicate la volta successiva che gli utenti cambiano le password. Gli utenti non sono costretti a modificare le proprie password esistenti, anche se le password esistenti non rispettano i criteri della policy aggiornata sulle password.
- Quando si imposta un periodo di scadenza della password, il periodo di scadenza viene applicato immediatamente. Ad esempio, un periodo di scadenza della password viene impostato a 90 giorni. In tal caso, la password scade per tutti gli utenti IAM la cui password attuale è più vecchia di 90 giorni. Gli utenti sono tenuti a modificare la propria password all'accesso successivo.

Non è possibile creare una "policy di esclusione" per bloccare un utente fuori dall'account dopo un numero specificato di tentativi di accesso non riusciti. Per una maggiore sicurezza, si consiglia di combinare una policy delle password complessa con l'autenticazione Multi-Factor Authentication (MFA). Per ulteriori informazioni sulla funzionalità MFA, consultare Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

Autorizzazioni necessarie per impostare una policy delle password

È necessario configurare le autorizzazioni per consentire a un'entità IAM (utente o ruolo) di visualizzare o modificare la policy delle password dell'account. È possibile includere le seguenti operazioni di policy della password in una policy IAM:

- `iam:GetAccountPasswordPolicy`: consente all'entità di visualizzare la policy delle password per il proprio account
- `iam:DeleteAccountPasswordPolicy`: consente all'entità di eliminare la policy delle password personalizzata per il proprio account e ripristinare la policy delle password di default

- `iam:UpdateAccountPasswordPolicy`: consente all'entità di creare o modificare la policy delle password personalizzata per il proprio account

La policy seguente consente l'accesso completo per visualizzare e modificare la policy delle password dell'account. Per ulteriori informazioni su come creare una policy IAM usando il documento di policy JSON di esempio, consulta the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessPasswordPolicy",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:DeleteAccountPasswordPolicy",
                "iam:UpdateAccountPasswordPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Per informazioni sulle autorizzazioni necessarie per modificare la password da parte di un utente IAM, consulta Consentire agli utenti IAM di cambiare le loro password.

Policy delle password predefinita

Se un amministratore non imposta una politica di password personalizzata, le password degli utenti IAM devono soddisfare la politica di AWS password predefinita.

La policy delle password predefinita applica le seguenti condizioni:

- Deve avere una lunghezza minima di 8 caratteri e massima di 128 caratteri
- Deve includere almeno tre dei seguenti tipi di caratteri: lettere maiuscole, lettere minuscole, numeri e caratteri non alfanumerici (`! @ # $ % ^ & * ( ) _ + - = [ ] { } | '`)
- Non essere identico al tuo Account AWS nome o indirizzo email
- Password che non scadono mai

Opzioni di policy delle password personalizzata

Quando si configura una policy delle password personalizzata per l'account, è possibile specificare le seguenti condizioni:

- Lunghezza minima della password: è possibile specificare un minimo di 6 caratteri e un massimo di 128 caratteri.
- Complessità della password: puoi selezionare una delle seguenti caselle di controllo per definire la complessità delle password dell'utente IAM:
  - Richiedi almeno una lettera maiuscola dall'alfabeto latino (A-Z)
  - Richiedi almeno una lettera minuscola dall'alfabeto latino (a-z)
  - Richiedere almeno un numero
  - Richiedere almeno un carattere non alfanumerico `! @ # $ % ^ & * ( ) _ + - = [ ] { } | '`
- Turn on password expiration (Abilita scadenza della password): puoi selezionare e specificare un minimo di 1 e un massimo di 1.095 giorni di validità delle password utente IAM dopo che sono state impostate. Ad esempio, se specifichi una scadenza di 90 giorni, ciò influisce immediatamente su tutti gli utenti. Dopo la modifica, gli utenti con password impostata da oltre 90 giorni dovranno impostarne una nuova quando accedono alla console. Gli utenti con password vecchie di 75-89 giorni ricevono un AWS Management Console avviso sulla scadenza della password. Gli utenti IAM possono modificare la password in qualsiasi momento se dispongono dell'autorizzazione. Quando impostano una nuova password, il periodo di scadenza per tale password ricomincia da capo. Un utente IAM può avere solo una password valida alla volta.
- La scadenza della password richiede la reimpostazione dell'amministratore: seleziona questa opzione per impedire agli utenti IAM di utilizzare il AWS Management Console per aggiornare le proprie password dopo la scadenza della password. Prima di selezionare questa opzione, verifica che nell' Account AWS sia presente più di un utente con le autorizzazioni amministrative per ripristinare le password degli utenti IAM. Gli amministratori che dispongono dell'autorizzazione `iam:UpdateLoginProfile` possono reimpostare le password degli utenti IAM. Gli utenti IAM che dispongono dell'autorizzazione `iam:ChangePassword` e di chiavi di accesso attive possono reimpostare autonomamente la propria password della console utente IAM a livello di programmazione. Se si deseleziona questa casella di controllo, gli utenti IAM con password scadute devono comunque impostare una nuova password prima di poter accedere alla AWS Management Console.
- Allow users to change their own password (Consenti agli utenti di modificare la propria password): puoi consentire a tutti gli utenti IAM nel tuo account di modificare autonomamente le proprie

password. Ciò consente agli utenti di accedere all'operazione `iam:ChangePassword` solo per il proprio utente e all'operazione `iam:GetAccountPasswordPolicy`. Questa opzione non associa una policy di autorizzazione a ciascun utente. Piuttosto, IAM applica le autorizzazioni a livello di account per tutti gli utenti. In alternativa, è possibile consentire solo ad alcuni utenti di gestire in autonomia le proprie password. A tale scopo, deseleziona questa casella di controllo. Per ulteriori informazioni sull'utilizzo di policy per limitare chi può gestire le password, consultare Consentire agli utenti IAM di cambiare le loro password.

- Impedisci il riutilizzo di una password: puoi impedire che gli utenti IAM riutilizzino un determinato numero di password precedenti. È possibile specificare un numero minimo di 1 e un numero massimo di 24 password precedenti che non possono essere ripetute.

Impostazione di una policy sulle password (Console)

Puoi utilizzare la AWS Management Console per creare, modificare o eliminare una politica di password personalizzata.

Per creare una policy delle password personalizzata (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
4. Scegli Custom (Personalizzato) per utilizzare una policy di password personalizzata.
5. Seleziona le opzioni che desideri applicare alla policy delle password e scegli Salva modifiche.
6. Conferma che desideri impostare la policy delle password personalizzata scegliendo Set custom (Imposta personalizzata).

Per modificare una policy delle password personalizzata (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
4. Seleziona le opzioni che desideri applicare alla policy delle password e scegli Salva modifiche.

5. Conferma che desideri impostare la policy delle password personalizzata scegliendo Set custom (Imposta personalizzata).

Eliminazione di una policy delle password personalizzata (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
4. Scegli IAM default (Predefinito IAM) per eliminare la policy di password personalizzata, quindi scegli Save changes (Salva modifiche).
5. Conferma che desideri impostare la policy delle password predefinita IAM scegliendo Set default (Imposta predefinita).

Impostazione di una policy sulle password (AWS CLI)

Puoi utilizzare il AWS Command Line Interface per impostare una politica in materia di password.

Per gestire la politica personalizzata in materia di password dell'account dal AWS CLI

Esegui i comandi seguenti:

- Per creare o modificare la policy delle password personalizzata: `aws iam update-account-password-policy`
- Per visualizzare la policy delle password: `aws iam get-account-password-policy`
- Per eliminare la policy delle password personalizzata: `aws iam delete-account-password-policy`

Impostazione di una politica in materia di password (AWS API)

È possibile utilizzare le operazioni AWS API per impostare una politica in materia di password.

Per gestire la politica personalizzata in materia di password dell'account dall' AWS API

Chiamare le operazioni seguenti:

- Per creare o modificare la policy delle password personalizzata: `UpdateAccountPasswordPolicy`

- Per visualizzare la policy delle password: `GetAccountPasswordPolicy`
- Per eliminare la policy delle password personalizzata: `DeleteAccountPasswordPolicy`

Gestione delle password per gli utenti IAM

Gli utenti IAM che utilizzano il AWS Management Console per lavorare con AWS le risorse devono disporre di una password per poter accedere. Puoi creare, modificare o eliminare una password di un utente IAM nel tuo account AWS .

Dopo aver assegnato una password a un utente, l'utente può accedere AWS Management Console utilizzando l'URL di accesso per il tuo account, che ha il seguente aspetto:

```
https://12-digit-AWS-account-ID or alias.signin.aws.amazon.com/console
```

Per ulteriori informazioni su come gli utenti IAM accedono a AWS Management Console, consulta Come accedere a AWS nella Guida per l'Accedi ad AWS utente.

Anche se gli utenti hanno le proprie password, hanno ancora bisogno delle autorizzazioni per accedere alle tue risorse AWS . Per impostazione predefinita, un utente non ha autorizzazioni. Per fornire agli utenti le autorizzazioni di cui hanno bisogno, assegna loro le policy o ai gruppi di appartenenza. Per ulteriori informazioni sulla creazione di utenti e gruppi, vedere Identità IAM (utenti, gruppi di utenti e ruoli). Per ulteriori informazioni sull'utilizzo di policy per impostare le autorizzazioni, vedere Modifica delle autorizzazioni per un utente IAM.

Puoi concedere le autorizzazioni agli utenti per modificare le loro password. Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password. Per informazioni su come gli utenti accedono alla pagina di accesso del tuo account, consulta Come accedere ad AWS nella Guida per l'utente di Accedi ad AWS .

Argomenti

- Creazione, modifica o eliminazione di una password dell'utente IAM (console)
- Creazione, modifica o eliminazione di una password utente IAM (AWS CLI)
- Creazione, modifica o eliminazione di una password utente IAM (API)AWS

Creazione, modifica o eliminazione di una password dell'utente IAM (console)

Puoi utilizzare il AWS Management Console per gestire le password per i tuoi utenti IAM.

Quando gli utenti lasciano l'organizzazione o non hanno più bisogno di AWS accedervi, è importante trovare le credenziali che stavano utilizzando e assicurarsi che non siano più operative. La soluzione ideale consiste nell'eliminare tutte le credenziali inutilizzate. Se l'utente dovesse averne bisogno in un secondo momento, potrai sempre ricrearle. Come minimo, dovresti modificare le credenziali, in modo che gli ex utenti non siano più in grado di poter accedere.

Come aggiungere una password per un utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Seleziona il nome dell'utente sul quale desideri creare la password.
4. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Enable console access (Abilita l'accesso alla console).
5. In Abilita l'accesso alla console, per la password della console, scegli se fare in modo che IAM generi una password o crei una password personalizzata:
   - Per fare in modo che IAM generi una password, scegli Password autogenerata
   - Per creare una password personalizzata, scegliere Custom password (Password personalizzata) e digitare la password.

   > **Note**
   >
   > La password creata deve essere conforme alla policy delle password dell'account.

6. Per richiedere all'utente di creare una nuova password al prossimo accesso. scegli User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo accesso). Quindi scegli Abilita l'accesso alla console.

   > **Important**
   >
   > Se si seleziona l'opzione User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo accesso), assicurati che l'utente disponga dell'autorizzazione per modificare la propria password. Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password.

7. Per visualizzare la password in modo da poterla condividere con l'utente, scegli Mostra nella finestra di dialogo della password della console.

> **Important**
>
> Per motivi di sicurezza, non è possibile accedere alla password dopo aver completato questa fase, ma è possibile creare una nuova password in qualsiasi momento.

Come cambiare la password per un utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Seleziona il nome dell'utente di cui desideri modificare la password.
4. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Manage console access (Gestisci l'accesso alla console).
5. In Gestisci l'accesso alla console, scegli Reimposta la password se non l'hai già selezionata. Se l'accesso alla console è disabilitato, non è richiesta alcuna password.
6. Per l'accesso alla console, scegli se fare in modo che IAM generi una password o crei una password personalizzata:
   - Per fare in modo che IAM generi una password, scegli Password autogenerata
   - Per creare una password personalizzata, scegliere Custom password (Password personalizzata) e digitare la password.

> **Note**
>
> La password creata deve essere conforme alla policy sulle password dell'account, se ne è stata impostata una.

7. Per richiedere all'utente di creare una nuova password al prossimo accesso. scegli User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo accesso).

> **Important**
>
> Se si seleziona l'opzione User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo accesso), assicurati che l'utente disponga dell'autorizzazione per modificare la propria password. Per ulteriori informazioni, consulta Consentire agli utenti IAM di cambiare le loro password.

8. Per revocare le sessioni di console attive dell'utente, scegli Revoca sessioni di console attive. Quindi, scegliere Apply (Applica).

   Quando revochi le sessioni attive della console per un utente, IAM attribuisce all'utente una nuova policy in linea che nega tutte le autorizzazioni a tutte le azioni. Include una condizione che applica le restrizioni solo se la sessione è stata creata prima del momento in cui si revocano le autorizzazioni, nonché dopo circa 30 secondi dal futuro. Se l'utente crea una nuova sessione dopo aver revocato le autorizzazioni, la politica di negazione non si applica a quell'utente.
   Se un utente revoca le proprie sessioni di console attive utilizzando questo metodo, verrà immediatamente disconnesso da. AWS Management Console

   > **Important**
   >
   > Per revocare correttamente le sessioni attive della console per un utente, è necessario disporre dell'`PutUserPolicy`autorizzazione dell'utente. Ciò consente di allegare la politica `AWSRevokeOlderSessions` in linea all'utente.

9. Per visualizzare la password in modo da poterla condividere con l'utente, scegli Mostra nella finestra di dialogo della password della console.

   > **Important**
   >
   > Per motivi di sicurezza, non è possibile accedere alla password dopo aver completato questa fase, ma è possibile creare una nuova password in qualsiasi momento.

Come eliminare (disabilitare) una password dell'utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, seleziona Utenti.
3. Seleziona il nome dell'utente di cui desideri eliminare la password.
4. Scegli la scheda Security credentials (Credenziali di sicurezza), quindi in Console sign-in (Accesso alla console), scegli Manage console access (Gestisci l'accesso alla console).
5. In Gestisci l'accesso alla console, scegli Disabilita l'accesso alla console se non è già selezionato. Se l'accesso alla console è disabilitato, non è richiesta alcuna password.
6. Per revocare le sessioni di console attive dell'utente, scegli Revoca sessioni di console attive. Quindi scegli Disabilita l'accesso.

> **Important**
>
> Per revocare correttamente le sessioni di console attive per un utente, devi disporre dell'`PutUserPolicy`autorizzazione per l'utente. Ciò consente di allegare la politica `AWSRevokeOlderSessions` in linea all'utente.

Quando revochi le sessioni di console attive per un utente, IAM incorpora una nuova policy in linea nell'utente IAM che nega tutte le autorizzazioni a tutte le azioni. Include una condizione che applica le restrizioni solo se la sessione è stata creata prima del momento in cui si revocano le autorizzazioni, nonché dopo circa 30 secondi dal futuro. Se l'utente crea una nuova sessione dopo aver revocato le autorizzazioni, la politica di negazione non si applica a quell'utente.
Se un utente revoca le proprie sessioni di console attive utilizzando questo metodo, verrà immediatamente disconnesso da. AWS Management Console

> **Important**
>
> Puoi impedire a un utente IAM di accedere a AWS Management Console rimuovendo la sua password. Ciò impedisce loro di accedere AWS Management Console utilizzando le proprie credenziali di accesso. Non modifica le autorizzazioni né impedisce l'accesso alla console utilizzando un ruolo assunto. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l' AWS CLI accesso tramite Tools for Windows PowerShell, AWS API o AWS Console Mobile Application.

Creazione, modifica o eliminazione di una password utente IAM (AWS CLI)

Puoi utilizzare l' AWS CLI API per gestire le password per i tuoi utenti IAM.

Per creare una password (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, esegui questo comando: aws iam get-login-profile
2. Per creare una password, esegui questo comando: aws iam create-login-profile

Per modificare la password di un utente (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, esegui questo comando: aws iam get-login-profile
2. Per modificare una password, esegui questo comando: aws iam update-login-profile

Per eliminare (disabilitare) una password utente (AWS CLI)

1. (Facoltativo) Per determinare se un utente dispone di una password, esegui questo comando: aws iam get-login-profile
2. (Facoltativo) Per determinare quando una password è stata utilizzata per l'ultima volta, eseguire questo comando: aws iam get-user
3. Per eliminare una password, esegui questo comando: aws iam delete-login-profile

> **Important**
>
> Quando elimini una password dell'utente, l'utente non può più accedere alla AWS Management Console. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l'accesso tramite AWS CLI le chiamate di funzione Tools for Windows PowerShell o AWS API. Quando utilizzi Tools for Windows o PowerShell l' AWS CLI AWS API per eliminare un utente dal tuo Account AWS, devi prima eliminare la password utilizzando questa operazione. Per ulteriori informazioni, consulta Eliminazione di un utente IAM (AWS CLI).

Per revocare le sessioni attive della console di un utente prima di un orario specificato ()AWS CLI

1. Per incorporare una policy in linea che revochi le sessioni di console attive di un utente IAM prima di un orario specificato, usa la seguente policy in linea ed esegui questo comando: aws iam put-user-policy

   Questa politica in linea nega tutte le autorizzazioni e include la chiave di condizione. `aws: TokenIssue Ora` Revoca le sessioni di console attive dell'utente prima del tempo specificato nell'`Condition`elemento della politica in linea. Sostituisci il valore della chiave della `aws:TokenIssueTime` condizione con il tuo valore.

   ```
   {
     "Version": "2012-10-17",
     "Statement": {
       "Effect": "Deny",
       "Action": "*",
       "Resource": "*",
       "Condition": {
         "DateLessThan": {
           "aws:TokenIssueTime": "2014-05-07T23:47:00Z"
         }
       }
     }
   }
   ```

2. (Facoltativo) Per elencare i nomi delle politiche in linea incorporate nell'utente IAM, esegui questo comando: aws iam list-user-policies
3. (Facoltativo) Per visualizzare la policy in linea denominata incorporata nell'utente IAM, esegui questo comando: aws iam get-user-policy

Creazione, modifica o eliminazione di una password utente IAM (API)AWS

Puoi utilizzare l' AWS API per gestire le password per i tuoi utenti IAM.

Per creare una password (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, chiamate questa operazione: GetLoginProfile
2. Per creare una password, chiamate questa operazione: CreateLoginProfile

Per modificare la password di un utente (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, richiamate questa operazione: GetLoginProfile
2. Per modificare una password, chiamate questa operazione: UpdateLoginProfile

Per eliminare (disabilitare) la password di un utente (AWS API)

1. (Facoltativo) Per determinare se un utente dispone di una password, esegui questo comando: GetLoginProfile
2. (Facoltativo) Per determinare quando è stata utilizzata l'ultima volta una password, esegui questo comando: GetUser
3. Per eliminare una password, esegui questo comando: DeleteLoginProfile

> **Important**
>
> Quando elimini una password dell'utente, l'utente non può più accedere alla AWS Management Console. Se l'utente dispone di chiavi di accesso attive, queste continuano a funzionare e consentono l'accesso tramite AWS CLI le chiamate di funzione Tools for Windows PowerShell o AWS API. Quando utilizzi Tools for Windows o PowerShell l' AWS CLI AWS API per eliminare un utente dal tuo Account AWS, devi prima eliminare la password utilizzando questa operazione. Per ulteriori informazioni, consulta Eliminazione di un utente IAM (AWS CLI).

Per revocare le sessioni di console attive di un utente prima di un orario specificato (API)AWS

1. Per incorporare una policy in linea che revochi le sessioni di console attive di un utente IAM prima di un orario specificato, utilizza la seguente policy in linea ed esegui questo comando: PutUserPolicy

   Questa politica in linea nega tutte le autorizzazioni e include la chiave di condizione. `aws:TokenIssue Ora` Revoca le sessioni di console attive dell'utente prima del tempo specificato nell'`Condition`elemento della politica in linea. Sostituisci il valore della chiave della `aws:TokenIssueTime` condizione con il tuo valore.

   ```
   {
   ```

```
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {
        "aws:TokenIssueTime": "2014-05-07T23:47:00Z"
      }
    }
  }
}
```

2. (Facoltativo) Per elencare i nomi delle politiche in linea incorporate nell'utente IAM, esegui questo comando: ListUserPolicies
3. (Facoltativo) Per visualizzare la policy in linea denominata incorporata nell'utente IAM, esegui questo comando: GetUserPolicy

Consentire agli utenti IAM di cambiare le loro password

> **Note**
>
> Per modificare le password, gli utenti con identità federate utilizzeranno il processo definito dal proprio gestore di identità. Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee.

Puoi concedere agli utenti IAM l'autorizzazione per modificare le password di accesso alla AWS Management Console. Ci sono due modi per farlo:

- Consenti a tutti gli utenti IAM nell'account di cambiare le loro password.
- Consentire solo agli utenti IAM selezionati di cambiare le loro password. In questo scenario, è possibile disattivare l'opzione di modifica della password per tutti gli utenti e utilizzare una policy IAM per concedere autorizzazioni solo ad alcuni utenti. Questo approccio consente a questi utenti di modificare le proprie password e, facoltativamente, altre credenziali, come le proprie chiavi di accesso.

> **Important**
>
> Consigliamo di impostare una policy delle password personalizzata che richieda agli utenti IAM di creare password complesse.

Per consentire a tutti gli utenti IAM di cambiare le loro password

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, fai clic su Impostazioni account.
3. Nella sezione Password policy (Policy delle password), scegli Edit (Modifica).
4. Scegli Custom (Personalizzato) per utilizzare una policy di password personalizzata.
5. Seleziona Allow users to change their own password (Consenti agli utenti di modificare la propria password), quindi scegli Save changes (Salva modifiche). Ciò consente a tutti gli utenti nell'account di accedere `iam:ChangePassword` all'operazione solo per il proprio utente e all'operazione `iam:GetAccountPasswordPolicy`.
6. Fornisci agli utenti le seguenti istruzioni per modificare le password: Come un utente IAM può modificare la propria password.

Per informazioni sui comandi Tools for Windows PowerShell e API che puoi utilizzare per modificare la politica in materia di password dell'account (che include la possibilità che tutti gli utenti cambino le proprie password), consulta. AWS CLIImpostazione di una policy sulle password (AWS CLI)

Per consentire a utenti IAM selezionati di cambiare le loro password.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, fai clic su Impostazioni account.
3. Nella sezione Impostazioni account, assicurati che l'opzione Consenti a tutti gli utenti di cambiare la propria password non sia selezionata. Se questa casella di controllo è selezionata, tutti gli utenti possono modificare le password. (Consulta la procedura precedente.)
4. Crea gli utenti che dovrebbero essere autorizzati a modificare la propria password, se non esistono ancora. Per informazioni dettagliate, vedi Creare un utente IAM nel tuo Account AWS.

5. (Facoltativo) Crea un gruppo IAM per gli utenti che possono modificare le loro password e aggiungi gli utenti dalla fase precedente a tale gruppo. Per informazioni dettagliate, vedi Gestione dei gruppi di utenti IAM.
6. Assegnare la policy seguente al gruppo. Per ulteriori informazioni, consulta Gestione di policy IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:GetAccountPasswordPolicy",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:ChangePassword",
      "Resource": "arn:aws:iam::*:user/${aws:username}"
    }
  ]
}
```

   Questa policy concede l'accesso all'ChangePasswordazione, che consente agli utenti di modificare solo le proprie password dalla console AWS CLI, dagli Strumenti per Windows PowerShell o dall'API. Concede inoltre l'accesso all'GetAccountPasswordPolicyazione, che consente all'utente di visualizzare la politica corrente in materia di password; questa autorizzazione è necessaria per consentire all'utente di visualizzare la politica sulla password dell'account nella pagina Modifica password. L'utente deve essere autorizzato a leggere la policy delle password corrente per assicurare che la password modificata soddisfi i requisiti della policy.
7. Fornisci agli utenti le seguenti istruzioni per modificare le password: Come un utente IAM può modificare la propria password.

Ulteriori informazioni

Per ulteriori informazioni sulla gestione delle credenziali, consultare i seguenti argomenti:

- Consentire agli utenti IAM di cambiare le loro password
- Gestione delle password degli utenti in AWS

- Impostazione di una policy delle password dell'account per utenti IAM
- Gestione di policy IAM
- Come un utente IAM può modificare la propria password

Come un utente IAM può modificare la propria password

Se ti è stata concessa l'autorizzazione a modificare la tua password utente IAM, puoi utilizzare una pagina speciale AWS Management Console per farlo. Puoi anche usare l' AWS API AWS CLI or.

Argomenti

- Autorizzazioni richieste
- Come gli utenti IAM possono cambiare le proprie password (console)
- In che modo gli utenti IAM modificano la propria password (AWS CLI o AWS API)

Autorizzazioni richieste

Per modificare la password del proprio utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente: AWS: consente agli utenti IAM di modificare la propria password della console nella pagina Credenziali di sicurezza.

Come gli utenti IAM possono cambiare le proprie password (console)

La procedura seguente descrive come gli utenti IAM possono utilizzare il AWS Management Console per modificare la propria password.

Come cambiare la propria password utente IAM (console)

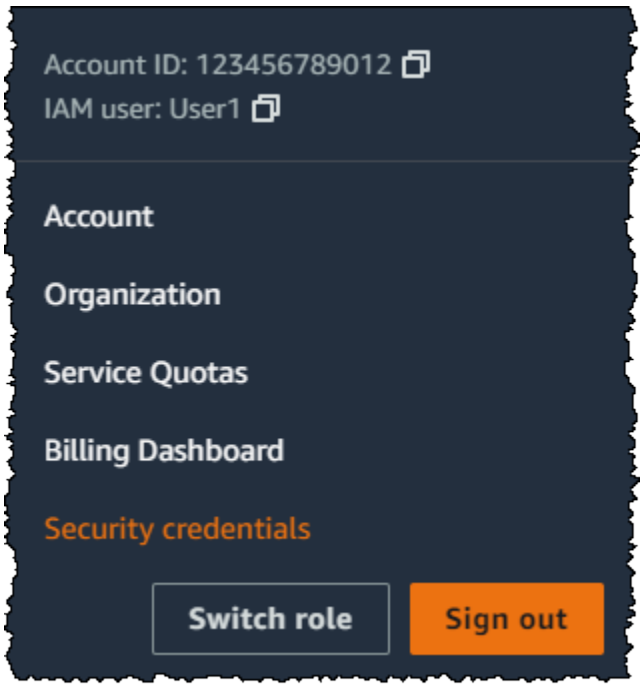
1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare

l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, seleziona Aggiorna password.
4. In Current password (Password corrente) digitare la password attuale. Digitare una nuova password per New password (Nuova password) e Confirm new password (Conferma nuova password). Quindi sceglie Aggiorna password.

> **Note**
>
> La nuova password deve soddisfare i requisiti della nuova policy delle password per l'account. Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM.

In che modo gli utenti IAM modificano la propria password (AWS CLI o AWS API)

La procedura seguente descrive come gli utenti IAM possono utilizzare l' AWS API AWS CLI or per modificare la propria password.

Per modificare la propria password IAM, utilizza i seguenti comandi:

- AWS CLI: `aws iam change-password`
- AWS API: `ChangePassword`

Gestione delle chiavi di accesso per gli utenti IAM

Follow us on Twitter

> **Important**
>
> Come best practice, utilizza credenziali di sicurezza temporanee (come i ruoli IAM) invece di creare credenziali a lungo termine come le chiavi di accesso. Prima di creare le chiavi di accesso, esamina le alternative alle chiavi di accesso a lungo termine.

Le chiavi di accesso sono credenziali a lungo termine per un utente IAM o l' Utente root dell'account AWS. Puoi utilizzare le chiavi di accesso per firmare le richieste programmatiche all' AWS API AWS CLI o (direttamente o utilizzando l' AWS SDK). Per ulteriori informazioni, consulta Firma AWS delle richieste API.

Le chiavi di accesso sono composte da due parti: un ID chiave di accesso (ad esempio `AKIAIOSFODNN7EXAMPLE`) e una chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). È necessario utilizzare sia l'ID chiave di accesso sia la chiave di accesso segreta insieme per autenticare le richieste dell'utente.

Se crei una coppia di chiavi di accesso, salva l'ID chiave di accesso e la chiave di accesso segreta in una posizione sicura. La chiave di accesso segreta è disponibile solo al momento della creazione. Se perdi la chiave di accesso segreta, è necessario eliminarla e crearne una nuova. Per ulteriori dettagli, consulta Reimpostazione delle password o delle chiavi di accesso perse o dimenticate per AWS.

È possibile avere al massimo due chiavi di accesso per utente.

Important

Gestisci le chiavi di accesso in modo sicuro. Non fornire le chiavi di accesso a parti non autorizzate, neppure per contribuire a trovare gli identificatori di account. Se lo facessi, daresti a qualcuno accesso permanente al tuo account.

I seguenti argomenti descrivono in dettaglio le attività di gestione associate alle chiavi di accesso.

Argomenti

Autorizzazioni necessarie per gestire le chiavi di accesso

Note

`iam:TagUser` è un'autorizzazione facoltativa per l'aggiunta e la modifica di descrizioni della chiave di accesso. Per ulteriori informazioni, consulta Tagging di utenti IAM

Per creare le chiavi di accesso per l'utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
```

```
                "iam:GetUser",
                "iam:ListAccessKeys",
                "iam:TagUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

Per aggiornare le chiavi di accesso per l'utente IAM, è necessario disporre delle autorizzazioni concesse dalla policy seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:GetAccessKeyLastUsed",
                "iam:GetUser",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:TagUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

Gestione delle chiavi di accesso (console)

Puoi utilizzare il AWS Management Console per gestire le chiavi di accesso di un utente IAM.

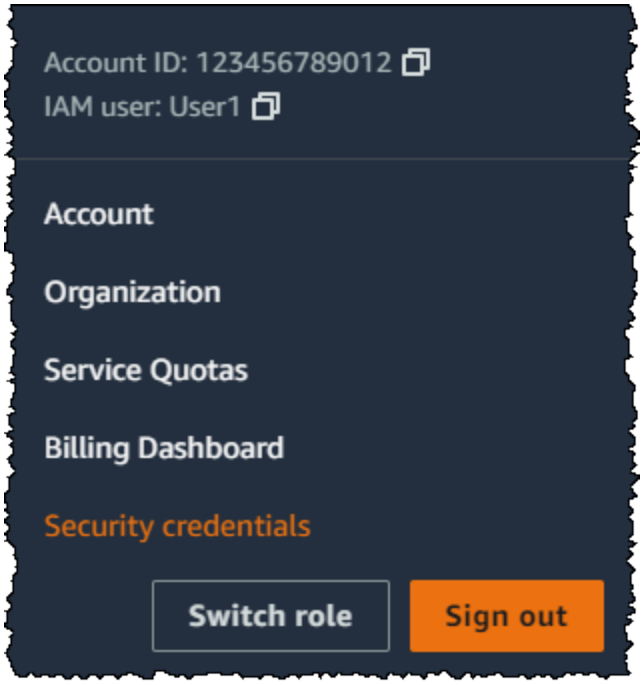
Per creare, modificare o eliminare le proprie chiavi di accesso (console)

1. Utilizza l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



Esegui una di queste operazioni:

Per creare una chiave di accesso

1. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Se dispone già di due chiavi di accesso, questo pulsante è disattivato e sarà necessario eliminare una chiave di accesso prima di crearne una nuova.

2. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), scegli il tuo caso d'uso per scoprire altre opzioni che possono aiutarti a evitare di creare una chiave di accesso a lungo termine. Se ritieni che il tuo caso d'uso richieda comunque una chiave di accesso, scegli Other (Altro) e poi Next (Successivo).
3. (Facoltativo) Imposta un valore del tag descrittivo per la chiave di accesso. Questo aggiunge una coppia chiave-valore di tag all'utente IAM. Ciò consente di identificare e aggiornare le chiavi di accesso in un secondo momento. La chiave di tag è impostata sull'ID della chiave di accesso. Il valore del tag è impostato sulla descrizione della chiave di accesso specificata. Al termine, scegli Create access key (Crea chiave di accesso).
4. Nella pagina Retrieve access keys (Recupera chiavi di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente o Download .csv file (Scarica il file .csv). Questa è la tua unica opportunità di salvare la chiave di accesso segreta. Dopo aver salvato la chiave di accesso segreta in una posizione sicura, scegli Done (Fatto).

Disattivazione di una chiave di accesso

- Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri disattivare, quindi scegli Actions (Operazioni) e poi Deactivate (Disattiva). Quando viene richiesta la conferma, scegliere Disattiva. Una chiave di accesso disattivata viene comunque conteggiata per il limite di due chiavi di accesso.

Attivazione di una chiave di accesso

- Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri attivare, quindi scegli Actions (Operazioni) e poi Activate (Attiva).

Eliminazione di una chiave di accesso quando non è più necessaria

- Nella sezione Access keys (Chiavi di accesso) individua la chiave che desideri eliminare, quindi scegli Actions (Operazioni) e poi Delete (Elimina). Segui le istruzioni nella finestra di dialogo prima per disattivare la chiave e poi conferma l'eliminazione. Si consiglia di verificare che la chiave di accesso non sia più in uso prima di eliminarla definitivamente.

Come creare, modificare o eliminare le chiavi di accesso di un altro utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Selezionare il nome dell'utente di cui si devono gestire le chiavi di accesso e selezionare la scheda Security credentials (Credenziali di sicurezza).
4. Nella sezione Access keys (Chiavi di accesso), procedere in uno dei seguenti modi:
   - Per creare una chiave di accesso, selezionare Create access key (Crea chiave di accesso). Se il pulsante è disattivato, dovrai eliminare una delle chiavi esistenti prima di poterne creare una nuova. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), esamina le best practice e le alternative. Scegli il tuo caso d'uso per scoprire altre opzioni che possono aiutarti a evitare di creare una chiave di accesso a lungo termine. Se ritieni che il tuo caso d'uso richieda comunque una chiave di accesso, scegli Other (Altro) e poi Next (Successivo). Nella pagina Retrieve access key (Recupera chiave di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente. Per salvare l'ID della chiave di accesso e la chiave di accesso segreta in un file `.csv` in una posizione sicura sul computer, seleziona il pulsante Download .csv file (Scarica file .csv). Quando crei una chiave di accesso per il tuo utente, la coppia di chiavi è attiva di default e può essere utilizzata immediatamente.
   - Per disattivare una chiave di accesso attiva, scegli Actions (Operazioni), quindi scegli Deactivate (Disattiva).
   - Per disattivare una chiave di accesso attiva, scegli Actions (Operazioni), quindi scegli Deactivate (Disattiva).
   - Per eliminare la chiave di accesso, scegli Actions (Operazioni) e poi Delete (Elimina). Segui le istruzioni nella finestra di dialogo prima per disattivare e poi per confermare l'eliminazione. Prima di eseguire questa operazione, AWS consiglia di disattivare la chiave e verificare che non sia più in uso. Quando usi il AWS Management Console, devi disattivare la chiave prima di eliminarla.

Come elencare le chiavi di accesso per un utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.

3. Selezionare il nome dell'utente, quindi selezionare la scheda Security credentials (Credenziali di sicurezza). Nella sezione Access keys (Chiavi di accesso), saranno visualizzate le chiavi di accesso dell'utente e lo stato di ciascuna chiave.

> **Note**
>
> Solo l'ID chiave di accesso dell'utente è visibile. La chiave di accesso segreta può essere recuperata solo al momento della creazione.

Come elencare gli ID chiave di accesso per più utenti IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Access key ID (ID chiave di accesso) alla tabella degli utenti mediante la procedura seguente:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Manage columns (Gestisci colonne) selezionare Access key ID (ID chiave di accesso).
   c. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.
4. La colonna Access key ID (ID chiave di accesso) mostra ogni ID chiave di accesso seguito dallo stato; ad esempio, 234782070278420732307623740 23 (Active) (Attivo) o 220937402396702370248434203 27 (Inactive) (Non attivo).

   È possibile utilizzare queste informazioni per visualizzare e copiare le chiavi di accesso per gli utenti con una o due chiavi di accesso. La colonna visualizza None (Nessuna) per gli utenti senza chiavi di accesso.

> **Note**
>
> Solo l'ID chiave di accesso dell'utente e il suo stato sono visibili. La chiave di accesso segreta può essere recuperata solo al momento della creazione.

Come determinare quale utente IAM possiede una determinata chiave di accesso (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Nella casella di ricerca digitare o incollare l'ID chiave di accesso dell'utente che si desidera individuare.
4. Se necessario, aggiungere la colonna Access key ID (ID chiave di accesso) alla tabella degli utenti mediante la procedura seguente:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Manage columns (Gestisci colonne) selezionare Access key ID (ID chiave di accesso).
   c. Selezionare Close (Chiudi) per tornare all'elenco di utenti e confermare che l'utente filtrato è proprietario della chiave di accesso specificata.

Gestione delle chiavi di accesso (AWS CLI)

Per gestire le chiavi di accesso utente IAM da AWS CLI, esegui i seguenti comandi.

- Per creare una chiave di accesso: `aws iam create-access-key`
- Attivazione o disattivazione di una chiave di accesso: `aws iam update-access-key`
- Per elencare le chiavi di accesso di un utente: `aws iam list-access-keys`
- Per determinare la data/ora più recente di utilizzo di una chiave di accesso: `aws iam get-access-key-last-used`
- Per eliminare una chiave di accesso: `aws iam delete-access-key`

Gestione delle chiavi di accesso (AWS API)

Per gestire le chiavi di accesso di un utente IAM dall' AWS API, richiama le seguenti operazioni.

- Per creare una chiave di accesso: `CreateAccessKey`
- Attivazione o disattivazione di una chiave di accesso: `UpdateAccessKey`
- Per elencare le chiavi di accesso di un utente: `ListAccessKeys`

- Per determinare la data/ora più recente di utilizzo di una chiave di accesso: `GetAccessKeyLastUsed`
- Per eliminare una chiave di accesso: `DeleteAccessKey`

Aggiornamento delle chiavi di accesso

Come best practice di sicurezza, è consigliabile aggiornare le chiavi di accesso degli utenti IAM all'occorrenza, ad esempio quando un dipendente lascia l'azienda. Gli utenti IAM possono aggiornare le proprie chiavi di accesso se dispongono delle autorizzazioni necessarie.

Per informazioni dettagliate su come concedere agli utenti IAM le autorizzazioni per aggiornare le proprie chiavi di accesso, consulta la pagina AWS: consente agli utenti IAM di gestire la propria password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza. Inoltre, è possibile applicare all'account una policy delle password per richiedere che tutti gli utenti IAM aggiornino periodicamente le loro password e con quale frequenza. Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM.

Argomenti

- Aggiornamento delle chiavi di accesso dell'utente IAM (console)
- Aggiornamento delle chiavi di accesso (AWS CLI)
- Aggiornamento delle chiavi di accesso (AWS API)

Aggiornamento delle chiavi di accesso dell'utente IAM (console)

È possibile aggiornare le chiavi di accesso dalla AWS Management Console.

Per aggiornare le chiavi di accesso per un utente IAM senza interrompere le applicazioni (console)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso.
    a. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
    b. Nel pannello di navigazione, seleziona Utenti.
    c. Selezionare il nome dell'utente, quindi selezionare la scheda Security credentials (Credenziali di sicurezza).

d. Nella sezione Chiavi di accesso, scegliere Crea chiave di accesso. Sulla pagina Access key best practices & alternatives (Best practice e alternative per le chiavi di accesso), scegli Other (Altro), quindi scegli Next (Successivo).

e. (Facoltativo) Imposta un valore del tag di descrizione per la chiave di accesso per aggiungere una coppia chiave-valore del tag a questo utente IAM. Ciò consente di identificare e aggiornare le chiavi di accesso in un secondo momento. La chiave di tag è impostata sull'ID della chiave di accesso. Il valore del tag è impostato sulla descrizione della chiave di accesso specificata. Al termine, scegli Create access key (Crea chiave di accesso).

f. Nella pagina Retrieve access keys (Recupera chiavi di accesso), scegli Show (Mostra) per rivelare il valore della chiave di accesso segreta dell'utente o Download .csv file (Scarica il file .csv). Questa è la tua unica opportunità di salvare la chiave di accesso segreta. Dopo aver salvato la chiave di accesso segreta in una posizione sicura, scegli Done (Fatto).

   Quando crei una chiave di accesso per il tuo utente, la coppia di chiavi è attiva di default e può essere utilizzata immediatamente. A questo punto, l'utente dispone di due chiavi di accesso attive.

2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.
3. Determina se la prima chiave di accesso è ancora in uso consultando la colonna Last used (Ultimo utilizzo) della chiave di accesso più vecchia. Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.
4. Anche se il valore della colonna Last used (Ultimo utilizzo) indica che la vecchia chiave non è mai stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Al contrario, seleziona Actions (Azioni) e poi Deactivate (Disattiva) per disattivare la prima chiave di accesso.
5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, puoi riattivare la prima chiave di accesso. Quindi, tornare a Step 3 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.
6. Dopo aver atteso un periodo di tempo per avere la certezza che tutte le applicazioni e gli strumenti siano stati aggiornati, è possibile eliminare la prima chiave di accesso:

   a. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

b. Nel pannello di navigazione, seleziona Utenti.

c. Selezionare il nome dell'utente, quindi selezionare la scheda Security credentials (Credenziali di sicurezza).

d. Nella sezione Access keys (Chiavi di accesso) individua la chiave di accesso che desideri eliminare, quindi scegli Actions (Operazioni) e poi Delete (Elimina). Segui le istruzioni nella finestra di dialogo prima per disattivare la chiave e poi per confermare l'eliminazione.

Per determinare quali chiavi di accesso devono essere aggiornate o eliminate (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Access key age (Durata chiave di accesso) alla tabella degli utenti mediante la procedura seguente:

   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).

   b. In Manage columns (Gestisci colonne) selezionare Access key age (Durata chiave di accesso).

   c. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.
4. La colonna Access key age (Durata chiave di accesso) mostra il numero di giorni trascorsi dalla creazione della più vecchia chiave di accesso attiva. È possibile utilizzare queste informazioni per trovare gli utenti per i quali potrebbe essere necessario aggiornare o eliminare le chiavi di accesso. La colonna visualizza None (Nessuna) per gli utenti senza chiavi di accesso.

Aggiornamento delle chiavi di accesso (AWS CLI)

È possibile aggiornare le chiavi di accesso dalla AWS Command Line Interface.

Per aggiornare le chiavi di accesso senza interrompere le applicazioni (AWS CLI)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso, che è attiva per default. Esegui il comando seguente:

   - `aws iam create-access-key`

A questo punto, l'utente dispone di due chiavi di accesso attive.

2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.
3. Determinare se la prima chiave di accesso è ancora in uso utilizzando questo comando:

   - `aws iam get-access-key-last-used`

   Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.
4. Anche se la fase Step 3 indica che la vecchia chiave non è stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Al contrario, modificare lo stato della prima chiave di accesso in `Inactive` utilizzando questo comando:

   - `aws iam update-access-key`
5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, è possibile ripristinare lo stato `Active` per riattivare la prima chiave di accesso. Quindi, tornare alla fase Step 2 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.
6. Dopo aver atteso un periodo di tempo per avere la certezza che tutte le applicazioni e gli strumenti siano stati aggiornati, è possibile eliminare la prima chiave di accesso con questo comando:

   - `aws iam delete-access-key`

Aggiornamento delle chiavi di accesso (AWS API)

È possibile aggiornare le chiavi di accesso utilizzando l' AWS API.

Per aggiornare le chiavi di accesso senza interrompere le applicazioni (AWS API)

1. Mentre la prima chiave di accesso è ancora attiva, creare una seconda chiave di accesso, che è attiva per default. Chiamare l'operazione seguente:

   - `CreateAccessKey`

   A questo punto, l'utente dispone di due chiavi di accesso attive.
2. Aggiornare tutte le applicazioni e gli strumenti in modo che utilizzino la nuova chiave di accesso.

3. Determinare se la prima chiave di accesso è ancora in uso chiamando questa operazione:

   - `GetAccessKeyLastUsed`

   Un approccio è aspettare diversi giorni e quindi verificare se la vecchia chiave di accesso sia stata utilizzata prima di procedere.

4. Anche se la fase Step 3 indica che la vecchia chiave non è stata utilizzata, è consigliabile non eliminare immediatamente la prima chiave di accesso. Al contrario, modificare lo stato della prima chiave di accesso in `Inactive` chiamando questa operazione:

   - `UpdateAccessKey`

5. Utilizzare solo la nuova chiave di accesso per verificare che le applicazioni funzionino. Tutte le applicazioni e gli strumenti che utilizzano ancora la chiave di accesso originale smetteranno di funzionare a questo punto perché non hanno più accesso alle AWS risorse. Se questo è il caso, è possibile ripristinare lo stato `Active` per riattivare la prima chiave di accesso. Quindi, tornare alla fase Step 2 e aggiornare l'applicazione in modo che utilizzi la nuova chiave.

6. Dopo aver atteso un periodo di tempo per avere la certezza che tutte le applicazioni e gli strumenti siano stati aggiornati, è possibile eliminare la prima chiave di accesso chiamando questa operazione:

   - `DeleteAccessKey`

Protezione delle chiavi di accesso

Chiunque disponga delle tue chiavi di accesso ha lo stesso livello di accesso alle tue AWS risorse che hai tu. Di conseguenza, AWS fa di tutto per proteggere le vostre chiavi di accesso e, in linea con il nostro modello di responsabilità condivisa, dovreste farlo anche voi.

Espandi le seguenti sezioni per ulteriori informazioni su come proteggere le chiavi di accesso.

> **Note**
>
> La tua organizzazione può avere policy e requisiti di sicurezza differenti rispetto a quelli descritti in questo argomento. I suggerimenti qui forniti sono destinati a essere linee guida generali.

Rimuovi (o non genera) le chiavi di accesso Utente root dell'account AWS

Uno dei modi migliori per proteggere il tuo account è non disporre di chiavi di accesso dell' Utente root dell'account AWS. A meno che non necessiti di disporre delle chiavi di accesso dell'utente root (il che è raro), è consigliabile non generarle. Crea invece un utente amministrativo AWS IAM Identity Center per le attività amministrative quotidiane. Per informazioni su come creare un utente amministrativo in IAM Identity Center, consulta la Guida introduttiva alla IAM Identity Center User Guide.

Se già disponi di chiavi di accesso dell'utente root per il tuo account, ti consigliamo di attenerti alle seguenti indicazioni: trova i punti nelle applicazioni in cui stai attualmente utilizzando le chiavi di accesso (se presenti) e sostituisci le chiavi di accesso dell'utente root con le chiavi di accesso dell'utente IAM. Quindi disabilita e rimuovi le chiavi di accesso dell'utente root. Per ulteriori informazioni sull'aggiornamento delle chiavi di accesso, consulta la pagina Aggiornamento delle chiavi di accesso

Utilizzo di credenziali di sicurezza temporanee (ruoli IAM) al posto delle chiavi di accesso a lungo termine

In molti scenari, non è necessaria una chiave di accesso a lungo termine a validità illimitata (come accade invece per gli utenti IAM). Al contrario, è possibile creare ruoli IAM e generare credenziali di sicurezza temporanee. Tali credenziali sono composte dall'ID della chiave di accesso e dalla chiave di accesso segreta, ma includono anche un token di sicurezza che ne indica la scadenza.

Le chiavi di accesso a lungo termine, ad esempio quelle associate a utenti IAM ed all'utente root, rimangono valide finché non vengono revocate manualmente. Tuttavia, le credenziali di sicurezza temporanee ottenute tramite i ruoli IAM e altre funzionalità di IAM AWS Security Token Service scadono dopo un breve periodo di tempo. Utilizza le credenziali di sicurezza temporanee per ridurre i rischi in caso di esposizione accidentale delle credenziali.

Utilizzare un ruolo IAM e le credenziali di sicurezza temporanee in questi scenari:

- Hai un'applicazione o AWS CLI degli script in esecuzione su un'istanza Amazon EC2. Non utilizzare le chiavi di accesso direttamente nell'applicazione. Non passare le chiavi di accesso all'applicazione, incorporarle nell'applicazione o lasciare che l'applicazione legga una chiave da qualsiasi origine. Al contrario, definisci un ruolo IAM con le autorizzazioni appropriate per l'applicazione e avvia l'istanza Amazon Elastic Compute Cloud (Amazon EC2) con i ruoli per EC2. In questo modo viene associato un ruolo IAM all'istanza Amazon EC2. Questa pratica, inoltre, consente all'applicazione di ottenere credenziali di sicurezza temporanee, che a sua volta può

utilizzare per effettuare chiamate a livello di programmazione ad AWS. Gli AWS SDK e AWS Command Line Interface (AWS CLI) possono ottenere automaticamente credenziali temporanee dal ruolo.

- Devi concedere l'accesso tra account. Utilizzare un ruolo IAM per stabilire l'attendibilità tra gli account, quindi concedere agli utenti di un account autorizzazioni limitate per accedere all'account attendibile. Per ulteriori informazioni, consulta Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.
- Hai a disposizione un'app mobile. Non integrare le chiavi di accesso con l'app, anche nell'archiviazione crittografata. Al contrario, utilizzare Amazon Cognito per la gestione dell'identità degli utenti nell'applicazione. Questo servizio consente di autenticare gli utenti utilizzando Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). È quindi possibile utilizzare il provider di credenziali Amazon Cognito per gestire le credenziali che l'app usa per le richieste ad AWS.
- Vuoi unirti a SAML 2.0 AWS e la tua organizzazione supporta SAML 2.0. Se si lavora per un'organizzazione che dispone di un provider di identità che supporta SAML 2.0, configurare il provider per l'utilizzo di SAML. Puoi utilizzare SAML per scambiare informazioni di autenticazione AWS e recuperare un set di credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta Federazione SAML 2.0.
- Vuoi eseguire la federazione AWS e la tua organizzazione dispone di un archivio di identità locale. Se gli utenti possono autenticarsi all'interno dell'organizzazione, è possibile scrivere un'applicazione in grado di emettere loro credenziali di sicurezza temporanee per l'accesso alle risorse. AWS Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

> **Note**
>
> Stai usando un'istanza Amazon EC2 con un'applicazione che richiede l'accesso programmatico alle risorse? AWS In tal caso, utilizza i ruoli IAM per EC2.

Gestione corretta delle chiavi di accesso dell'utente IAM

Se devi creare chiavi di accesso per l'accesso programmatico AWS, creale per gli utenti IAM, concedendo agli utenti solo le autorizzazioni di cui hanno bisogno.

Osserva queste precauzioni per proteggere le chiavi di accesso degli utenti IAM:

- Non incorporare le chiavi di accesso direttamente nel codice. Gli SDK AWS e gli Strumenti da linea di comando AWS consentono di collocare le chiavi di accesso in posizioni note in modo da evitare di conservarle nel codice.

  Colloca le chiavi di accesso in una delle posizioni seguenti:

  - Il AWS file delle credenziali. Gli AWS SDK utilizzano AWS CLI automaticamente le credenziali archiviate nel file delle AWS credenziali.

    Per informazioni sull'utilizzo del file delle AWS credenziali, consulta la documentazione del tuo SDK. Gli esempi includono Set AWS Credentials and Region nella AWS SDK for Java Developer Guide e i file di configurazione e credenziali nella Guida per l'utente.AWS Command Line Interface

    Per memorizzare le credenziali per AWS SDK for .NET and the AWS Tools for Windows PowerShell, ti consigliamo di utilizzare SDK Store. Per ulteriori informazioni, consulta Utilizzo dell'SDK Store nella Guida per gli sviluppatori di AWS SDK for .NET .
  - Variabili di ambiente. In un sistema multi-tenant, scegli le variabili di ambiente dell'utente e non le variabili di ambiente del sistema.

    Per ulteriori informazioni sull'utilizzo di variabili di ambiente per archiviare le credenziali, consultare la sezione Variabili di ambiente nella Guida per l'utente di AWS Command Line Interface .
- Utilizza chiavi di accesso diverse per applicazioni differenti. Esegui questa operazione in modo da isolare le autorizzazioni e revocare le chiavi di accesso per le singole applicazioni in caso una di esse venga esposta. Avere chiavi di accesso separate per applicazioni diverse genera anche voci distinte nei file di log AWS CloudTrail. Questa configurazione consente di determinare più facilmente quale applicazione ha eseguito azioni specifiche.
- Aggiorna le chiavi di accesso all'occorrenza. Se esiste il rischio che la chiave di accesso possa essere compromessa, aggiorna la chiave di accesso ed elimina quella precedente. Per maggiori dettagli, consulta Aggiornamento delle chiavi di accesso.
- Rimuovi le chiavi di accesso inutilizzate. Se un utente lascia l'organizzazione, rimuovere l'utente IAM corrispondente in modo che non possa più accedere alle risorse. Per scoprire quando è stata utilizzata l'ultima volta una chiave di accesso, utilizza l'`GetAccessKeyLastUsed`API (AWS CLI comando: `aws iam get-access-key-last-used`).
- Utilizza le credenziali temporanee e configura l'autenticazione a più fattori (MFA) per le operazioni API più sensibili. Con le policy IAM, è possibile specificare le operazioni API che un utente è autorizzato a chiamare. In alcuni casi, potresti richiedere la sicurezza aggiuntiva di

richiedere l'autenticazione degli utenti con AWS MFA prima di consentire loro di eseguire azioni particolarmente sensibili. Potrebbe ad esempio esserci una policy che permette a un utente di eseguire le operazioni `RunInstances`, `DescribeInstances` e `StopInstances` di Amazon EC2. Ma potresti voler limitare un'azione distruttiva come `TerminateInstances` e assicurarti che gli utenti possano eseguire tale azione solo se si autenticano con un dispositivo AWS MFA. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto da MFA.

Accedi all'app per dispositivi mobili utilizzando i tasti di accesso AWS

Puoi accedere a un set limitato di AWS servizi e funzionalità utilizzando l'app AWS mobile. L'app mobile ti aiuta a supportare la risposta agli incidenti mentre sei in viaggio. Per ulteriori informazioni e per scaricare l'app, consulta AWS Console Mobile Application.

È possibile accedere all'app per dispositivi mobili utilizzando la password della console o le chiavi di accesso. Come best practice, non utilizzare le chiavi di accesso dell'utente root. Ti consigliamo invece vivamente, oltre a utilizzare una password o un blocco biometrico sul tuo dispositivo mobile, di creare un utente IAM specifico per la gestione AWS delle risorse tramite l'app mobile. Se si perde il dispositivo mobile, è possibile rimuovere l'accesso dell'utente IAM.

Accesso mediante le chiavi di accesso (app per dispositivi mobili)

1. Apri l'app sul tuo dispositivo mobile.
2. Se questa è la prima volta che aggiungi un'identità al dispositivo, scegli Add an identity (Aggiungi un'identità) e scegli Access keys (Chiavi di accesso).

   Se hai già effettuato l'accesso utilizzando un'altra identità, scegli l'icona del menu e scegli Switch identity (Cambia identità). Quindi scegli Sign in as a different identity (Accedi come identità diversa) e quindi Access keys (Chiavi di accesso).
3. Nella pagina Access keys (Chiavi di accesso) immetti le informazioni nei campi.

   - ID chiave di accesso: immettere l'ID chiave di accesso.
   - Chiave di accesso segreta: inserire la chiave di accesso segreta.
   - Nome dell'identità: immettere il nome dell'identità che verrà visualizzata nell'applicazione per dispositivi mobili. Non è necessario che corrisponda al nome utente IAM.
   - PIN identità: creare un PIN (Personal Identification Number) da utilizzare per gli accessi futuri.

> **Note**
>
> Se abiliti la biometria per l'app AWS mobile, ti verrà richiesto di utilizzare l'impronta digitale o il riconoscimento facciale per la verifica anziché il PIN. Se la biometria restituisce un errore, potrebbe venire richiesto il PIN.

4. Scegliere Verify and add keys (Verifica e aggiungi chiavi).

   È ora possibile accedere a un set selezionato di risorse mediante l'app per dispositivi mobili.

Informazioni correlate

I seguenti argomenti forniscono indicazioni per la configurazione degli AWS SDK e l'utilizzo delle chiavi di accesso: AWS CLI

- Imposta AWS le credenziali e la regione nella Guida per gli sviluppatori AWS SDK for Java
- Utilizzo dell'SDK Store nella Guida per gli sviluppatori di AWS SDK for .NET .
- Specifica delle credenziali all'SDK nella Guida per gli sviluppatori di AWS SDK for PHP .
- Configurazione nella documentazione di Boto 3 (AWS SDK per Python)
- Utilizzo delle credenziali AWS nella Guida per l'utente di AWS Tools for Windows PowerShell
- File di configurazione e delle credenziali nella Guida per l'utente di AWS Command Line Interface .
- Concessione dell'accesso utilizzando un ruolo IAM nella Guida per gli sviluppatori di AWS SDK for .NET
- Configurazione dei ruoli IAM per Amazon EC2 nell'AWS SDK for Java 2.x

Audit delle chiavi di accesso

Puoi esaminare le chiavi di AWS accesso contenute nel codice per determinare se le chiavi provengono da un account di tua proprietà. Puoi passare l'ID di una chiave di accesso utilizzando il `aws sts get-access-key-info` AWS CLI comando o l'operazione `GetAccessKeyInfo` AWS API.

Le operazioni AWS CLI and AWS API restituiscono l'ID Account AWS a cui appartiene la chiave di accesso. Gli ID delle chiavi di accesso che iniziano con `AKIA` sono credenziali a lungo termine per un utente IAM o un Utente root dell'account AWS. Gli ID delle chiavi di accesso che iniziano con `ASIA`

sono credenziali temporanee create utilizzando AWS STS le operazioni. Se l'account nella risposta appartiene a te, puoi effettuare l'accesso come utente root e rivedere le chiavi di accesso dell'utente root. Quindi, puoi estrarre un report delle credenziali per scoprire quale utente IAM possiede le chiavi. Per sapere chi ha richiesto le credenziali temporanee per una chiave di `ASIA` accesso, visualizza gli AWS STS eventi nei tuoi CloudTrail registri.

Per motivi di sicurezza, puoi esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in. AWSÈ possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`.

Questa operazione non indica lo stato della chiave di accesso. La chiave potrebbe essere attiva, inattiva o eliminata. Le chiavi attive potrebbero non disporre delle autorizzazioni per eseguire un'operazione. Fornire una chiave di accesso eliminata potrebbe restituire un errore indicante che la chiave non esiste.

Reimpostazione delle password o delle chiavi di accesso perse o dimenticate per AWS

> **Important**
>
> Hai problemi ad accedere a? AWS Assicurati di essere nella pagina di accesso AWS corretta per il tuo tipo di utente. Se sei il Utente root dell'account AWS (proprietario dell'account), puoi accedere AWS utilizzando le credenziali che hai configurato quando hai creato il Account AWS. Se sei un utente IAM, l'amministratore dell'account può fornirti le credenziali che puoi utilizzare per accedere ad AWS. Se hai bisogno di richiedere assistenza, non utilizzare il link di feedback in questa pagina, poiché il modulo non AWS Support viene ricevuto dal team addetto alla AWS documentazione. Invece, nella pagina Contattaci, scegli Non riesci ancora ad accedere al tuo account AWS ? e scegli una delle opzioni di supporto disponibili.

Nella pagina di accesso principale, è necessario inserire il proprio indirizzo e-mail per accedere come utente root o l'ID account per accedere come utente IAM. È possibile fornire la password solo nella pagina di accesso corrispondente al tipo di utente. Per ulteriori informazioni, consulta Firma nella AWS Management Console.

Se ti trovi nella pagina di accesso corretta e perdi o dimentichi le password o le chiavi di accesso, non puoi recuperarle da IAM. Puoi invece reimpostarle usando i metodi seguenti:

- Utente root dell'account AWS password: se dimentichi la password dell'utente root, puoi reimpostarla da AWS Management Console. Per informazioni dettagliate, consulta the section called “Reimpostazione di una password dell'utente root persa o dimenticata” più avanti in questo argomento.
- Account AWS chiavi di accesso: se dimentichi le chiavi di accesso dell'account, puoi creare nuove chiavi di accesso senza disabilitare le chiavi di accesso esistenti. Se non stai usando le chiavi esistenti, puoi eliminarle. Per informazioni dettagliate, consulta Creazione di chiavi di accesso per l'utente root e Eliminazione di chiavi di accesso per l'utente root.
- Password utente IAM: se sei un utente IAM e dimentichi la password, devi chiedere all'amministratore di reimpostarla. Per ulteriori informazioni su come un amministratore può gestire la password, consulta Gestione delle password per gli utenti IAM.
- Chiavi di accesso utente IAM: se sei un utente IAM e dimentichi le chiavi di accesso, ti serviranno nuove chiavi di accesso. Se hai l'autorizzazione per creare le tue chiavi di accesso, puoi trovare le istruzioni per la creazione di nuove chiavi in Gestione delle chiavi di accesso (console). Se non hai le autorizzazioni necessarie, devi chiedere all'amministratore di creare nuove chiavi di accesso. Se stai ancora usando le vecchie chiavi, chiedi all'amministratore di non eliminarle. Per ulteriori informazioni su come un amministratore può gestire le chiavi di accesso, consulta Gestione delle chiavi di accesso per gli utenti IAM.

Utilizzo dell'autenticazione a più fattori (MFA) in AWS

Follow us on Twitter

Per una maggiore sicurezza, ti consigliamo di configurare l'autenticazione a più fattori (MFA) per proteggere AWS le tue risorse. Puoi abilitare l'MFA per gli utenti Utente root dell'account AWS e IAM. Quando abiliti MFA per l'utente root, questa impostazione influisce solo sulle credenziali dell'utente root. Gli utenti IAM nell'account sono identità distinte con proprie credenziali e ogni identità ha la propria configurazione MFA.

Puoi registrare fino a otto dispositivi MFA in qualsiasi combinazione dei tipi di MFA attualmente supportati con l' Utente root dell'account AWS e gli utenti IAM. Per ulteriori informazioni sui tipi di MFA supportati, consulta Tipi di MFA disponibili per gli utenti IAM. Con più dispositivi MFA, è necessario un solo dispositivo MFA per accedere AWS Management Console o creare una sessione tramite l' AWS CLI as quell'utente.

> **Note**
>
> Si consiglia di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWS Hai considerato l'utilizzo AWS IAM Identity Center? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli utenti un accesso Single Sign-On protetto da MFA a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Tipi di MFA disponibili per gli utenti IAM

La MFA aggiunge ulteriore sicurezza perché richiede agli utenti di fornire un'autenticazione unica da un meccanismo AWS MFA supportato oltre alle normali credenziali di accesso quando accedono a siti Web o servizi. AWS AWS supporta i seguenti tipi di MFA: passkey e chiavi di sicurezza, applicazioni di autenticazione virtuale e token TOTP hardware.

Passkey e chiavi di sicurezza

AWS Identity and Access Management supporta passkey e chiavi di sicurezza per MFA. In base agli standard FIDO, le passkey utilizzano la crittografia a chiave pubblica per fornire un'autenticazione forte e resistente al phishing, più sicura delle password. AWS supporta due tipi di passkey: passkey legate al dispositivo (chiavi di sicurezza) e passkey sincronizzate.

- Chiavi di sicurezza: si tratta di dispositivi fisici, come un YubiKey, utilizzati come secondo fattore di autenticazione.
- Passkey sincronizzate: come secondo fattore utilizzano gestori di credenziali di provider come Google, Apple, account Microsoft e servizi di terze parti come 1Password, Dashlane e Bitwarden.

Puoi utilizzare gli autenticatori biometrici integrati, come Touch ID su Apple MacBooks e il riconoscimento facciale Windows Hello sui PC, per sbloccare il gestore delle credenziali e accedere a. AWS Le passkey vengono create con il provider scelto utilizzando il PIN dell'impronta digitale, del viso o del dispositivo. Puoi sincronizzare le passkey tra i tuoi dispositivi per facilitare gli accessi e migliorare l'usabilità e la AWS recuperabilità.

FIDO Alliance mantiene un elenco di tutti i prodotti certificati FIDO compatibili con le specifiche FIDO. Una singola passkey o chiave di sicurezza supporta più account utente root e utenti IAM. Per ulteriori informazioni sull'abilitazione delle passkey e delle chiavi di sicurezza per un utente IAM, consulta. Abilitazione di una passkey o di una chiave di sicurezza (console)

Applicazioni di autenticazione virtuale

Un'applicazione di autenticazione virtuale viene eseguita su un telefono o altro dispositivo ed emula un dispositivo fisico. Le app di autenticazione virtuale implementano l'algoritmo TOTP (password monouso) e supportano più token su un singolo dispositivo. L'utente deve digitare un codice valido dal dispositivo quando richiesto durante l'accesso. Ogni token assegnato a un utente deve essere unico. Un utente non può digitare un codice dal token di un altro utente per l'autenticazione.

È consigliabile utilizzare un dispositivo MFA virtuale nell'attesa dell'approvazione di un acquisto hardware o della consegna del dispositivo hardware. Per un elenco di alcune app supportate che puoi utilizzare come dispositivi MFA virtuali, consulta Multi-Factor Authentication (MFA). Per istruzioni sulla configurazione di un dispositivo MFA virtuale per un utente IAM, consulta. Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console)

Token TOTP hardware

Un dispositivo hardware genera un codice numerico a sei cifre basato sull'algoritmo TOTP (Time-based One-Time Password). L'utente deve immettere un codice valido dal dispositivo su una seconda pagina Web durante la procedura di accesso. Ogni dispositivo MFA assegnato a un utente deve essere univoco. Per essere autenticati, gli utenti non possono digitare un codice generato dal dispositivo di un altro utente. Per informazioni sui dispositivi hardware MFA supportati, vedere Multi-Factor Authentication (MFA). Per istruzioni sulla configurazione di un token TOTP hardware per un utente IAM, consulta. Abilitazione di un token TOTP hardware (console)

Se desideri utilizzare un dispositivo MFA fisico, ti consigliamo di utilizzare le chiavi di sicurezza come alternativa ai dispositivi TOTP hardware. Le chiavi di sicurezza offrono i vantaggi dell'assenza di batteria, della resistenza al phishing e supportano più utenti root e IAM su un unico dispositivo per una maggiore sicurezza.

> **Note**
>
> Autenticazione MFA basata su SMS: AWS ha terminato il supporto per l'abilitazione dell'autenticazione a più fattori (MFA) con SMS. Consigliamo ai clienti con utenti IAM che utilizzano MFA basata su SMS di passare a uno dei seguenti metodi alternativi: passkey o

chiave di sicurezza, dispositivo MFA virtuale (basato su software) odispositivo MFA hardware. Puoi identificare gli utenti nel tuo account con un dispositivo MFA SMS assegnato. Per farlo, vai alla console IAM, scegli Users (Utenti) dal riquadro di navigazione e individua gli utenti con SMS nella colonna della tabella MFA.

Argomenti

Abilitazione dei dispositivi MFA per gli utenti in AWS

La procedura di configurazione dell'autenticazione MFA dipende dal tipo di dispositivo MFA che utilizzi.

Argomenti

Procedura generale per l'abilitazione dei dispositivi MFA

La seguente panoramica della procedura descrive come impostare e utilizzare MFA e fornisce collegamenti alle informazioni correlate.

> **Nota**
>
> Per ulteriori informazioni, puoi anche guardare questo video in lingua inglese, How to Setup AWS Multi-Factor Authentication (AWS MFA) and Budget Alerts.

1. Procurarsi un dispositivo MFA come uno dei seguenti. Puoi abilitare fino a otto dispositivi MFA per Utente root dell'account AWS utente IAM di qualsiasi combinazione dei seguenti tipi.
   - Un dispositivo MFA virtuale, ovvero un'app software conforme a RFC 6238, un algoritmo TOTP (password monouso temporanea) basato su standard. Puoi installare l'app su un telefono o su un altro dispositivo. Per un elenco di alcune delle app supportate che puoi utilizzare come dispositivi MFA virtuali, consulta la pagina Multi-Factor Authentication.
   - Una passkey o una chiave di sicurezza con una configurazione AWS supportata. FIDO Alliance mantiene un elenco di tutti i prodotti certificati FIDO compatibili con le specifiche FIDO.
   - Un dispositivo MFA basato su hardware di un provider terzo, ad esempio un dispositivo token. Questi token vengono utilizzati esclusivamente con. Account AWS Per ulteriori informazioni, consulta Abilitazione di un token TOTP hardware (console). Puoi utilizzare solo token con i loro token seed unici condivisi in modo sicuro. AWS I token seed sono chiavi segrete generate al momento della produzione dei token. I token acquistati da altre fonti non funzioneranno con IAM. Per garantire la compatibilità, è necessario acquistare il dispositivo hardware MFA da uno dei seguenti link: token OTP o scheda video OTP.
2. Abilitare il dispositivo MFA.
   - Token TOTP virtuali o hardware: puoi utilizzare AWS CLI comandi o operazioni AWS API per abilitare un dispositivo MFA virtuale per un utente IAM. Non è possibile abilitare un dispositivo MFA per il Utente root dell'account AWS AWS CLI, AWS API, Tools for Windows PowerShell o qualsiasi altro strumento da riga di comando. Tuttavia, è possibile utilizzare il AWS Management Console per abilitare un dispositivo MFA per l'utente root.
   - Passkey e chiavi di sicurezza: gli utenti root e gli utenti IAM con passkey o chiavi di sicurezza possono eseguire l'attivazione AWS Management Console solo dall' AWS CLI API o. AWS

   Per informazioni sull'abilitazione di ciascun tipo di dispositivo MFA, consultare le pagine seguenti:
   - Dispositivo MFA virtuale: Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console)
   - Passkey e chiavi di sicurezza: Abilitazione di una passkey o di una chiave di sicurezza (console)
   - Token TOTP hardware: Abilitazione di un token TOTP hardware (console)

3. Abilitazione di più dispositivi MFA (consigliato)
   - Ti consigliamo di abilitare più dispositivi MFA per gli utenti IAM del Utente root dell'account AWS tuo. Account AWS Ciò consente di aumentare la sicurezza degli Account AWS e semplificare la gestione dell'accesso agli utenti con privilegi elevati, come l' Utente root dell'account AWS.
   - Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Con più dispositivi MFA, è sufficiente un solo dispositivo MFA per accedere AWS Management Console o creare una sessione tramite l' AWS CLI as quell'utente. Per abilitare o disabilitare un dispositivo MFA aggiuntivo, un utente IAM deve prima autenticarsi con un dispositivo MFA esistente.
   - In caso di smarrimento, furto o inaccessibilità di un dispositivo MFA, è possibile utilizzare uno dei dispositivi MFA rimanenti per accedervi senza eseguire la Account AWS procedura di ripristino. Account AWS In caso di smarrimento o furto di un dispositivo MFA, consigliamo di dissociare il dispositivo dal principale IAM a cui era associato.
   - L'uso di più MFA consente ai dipendenti che si trovano in località geograficamente disperse o che lavorano in remoto di utilizzare l'MFA basata su hardware per accedere AWS senza dover coordinare lo scambio fisico di un singolo dispositivo hardware tra i dipendenti.
   - L'uso di dispositivi MFA aggiuntivi per i principali IAM consente di utilizzare uno o più MFA per l'uso quotidiano, mantenendo al contempo i dispositivi MFA fisici in un luogo fisico sicuro come un vault o una cassaforte per il backup e la ridondanza.
4. Utilizzare il dispositivo MFA quando si effettua l'accesso alle risorse AWS .
   - Passkey e chiavi di sicurezza: per accedere a un AWS sito Web, inserisci le tue credenziali e, a seconda del tipo di passkey che possiedi, tocca la chiave di sicurezza FIDO, inserisci un PIN del dispositivo o fornisci l'impronta digitale o il viso quando richiesto.
   - Dispositivi MFA virtuali e token TOTP hardware: per accedere a un AWS sito Web, è necessario un codice MFA dal dispositivo oltre al nome utente e alla password.

     Per accedere alle operazioni API protette da autenticazione MFA, è necessario quanto segue:
     - Un codice MFA
     - L'identificativo del dispositivo MFA (il numero di serie del dispositivo di un dispositivo fisico o l'ARN di un dispositivo virtuale definito in AWS)
     - I consueti ID chiave di accesso e chiave di accesso segreta

> **Note**
>
> - Non è possibile passare le informazioni MFA per una chiave di sicurezza FIDO alle operazioni AWS STS API per richiedere credenziali temporanee.
> - Non è possibile utilizzare AWS CLI comandi o operazioni AWS API per abilitare le chiavi di sicurezza FIDO.
> - Non è possibile utilizzare lo stesso nome per più di un dispositivo root o MFA IAM.

Per ulteriori informazioni, consulta Utilizzo di dispositivi MFA con la pagina di accesso IAM.

Abilitazione di una passkey o di una chiave di sicurezza (console)

Le passkey sono un tipo di dispositivo di autenticazione a più fattori (MFA) che puoi utilizzare per proteggere le tue risorse. AWS AWS supporta passkey sincronizzate e passkey legate al dispositivo, note anche come chiavi di sicurezza.

Le passkey sincronizzate consentono agli utenti IAM di accedere alle proprie credenziali di accesso FIDO su molti dei loro dispositivi, anche su quelli nuovi, senza dover registrare nuovamente tutti i dispositivi su ogni account. Le passkey sincronizzate includono gestori di credenziali proprietari come Google, Apple e Microsoft e gestori di credenziali di terze parti come 1Password, Dashlane e Bitwarden come secondo fattore. Puoi anche utilizzare la biometria sul dispositivo (ad esempio, TouchID, FaceID, Windows Hello) per sbloccare il gestore di credenziali scelto per utilizzare le passkey.

In alternativa, le password legate al dispositivo sono associate a una chiave di sicurezza FIDO che puoi collegare a una porta USB del computer e quindi toccare quando richiesto per completare in modo sicuro la procedura di accesso. Se utilizzi già una chiave di sicurezza FIDO con altri servizi e ha una configurazione AWS supportata (ad esempio, la serie YubiKey 5 di Yubico), puoi usarla anche con. AWS Altrimenti, è necessario acquistare una chiave di sicurezza FIDO se si desidera utilizzarla WebAuthn per l' AWS MFA in. Inoltre, le chiavi di sicurezza FIDO possono supportare più utenti IAM o root sullo stesso dispositivo, migliorandone l'utilità per la sicurezza degli account. Per specifiche e informazioni sull'acquisto per entrambi i tipi di dispositivo, consulta Multi-Factor Authentication.

Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Con più dispositivi MFA, è sufficiente un solo dispositivo MFA per accedere AWS Management Console o creare una sessione tramite

l' AWS CLI as quell'utente. Consigliamo di registrare più dispositivi MFA. Ad esempio, è possibile registrare un autenticatore integrato e anche una chiave di sicurezza da conservare in un luogo fisicamente sicuro. Se è impossibile utilizzare l'autenticatore integrato, si può utilizzare la chiave di sicurezza registrata. Per le applicazioni di autenticazione, consigliamo inoltre di abilitare le funzionalità di backup o sincronizzazione su cloud in tali app per evitare di perdere l'accesso all'account in caso di smarrimento o guasto del dispositivo che dispone delle app di autenticazione.

> **Note**
>
> Consigliamo di richiedere agli utenti di utilizzare credenziali temporanee per l'accesso a AWS. I tuoi utenti possono unirsi AWS a un provider di identità dove si autenticano con le credenziali aziendali e le configurazioni MFA. Per gestire l'accesso AWS e le applicazioni aziendali, ti consigliamo di utilizzare IAM Identity Center. Per ulteriori informazioni, consulta la Guida per l'utente di IAM Identity Center.

Argomenti

Autorizzazioni richieste

Per gestire una passkey FIDO per il tuo utente IAM proteggendo al contempo le azioni sensibili relative all'MFA, devi disporre delle autorizzazioni previste dalla seguente policy:

> **Note**
>
> I valori dell'ARN sono valori statici e non sono un indicatore del protocollo utilizzato per registrare l'autenticatore. U2F è obsoleto, quindi tutte le nuove implementazioni lo utilizzano. WebAuthn

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

Abilita una passkey o una chiave di sicurezza per il tuo utente IAM (console)

Puoi abilitare una passkey o una chiave di sicurezza per il tuo utente IAM AWS Management Console solo dall'API AWS CLI o AWS , non dall'API. Prima di poter abilitare una chiave di sicurezza, devi avere accesso fisico al dispositivo.

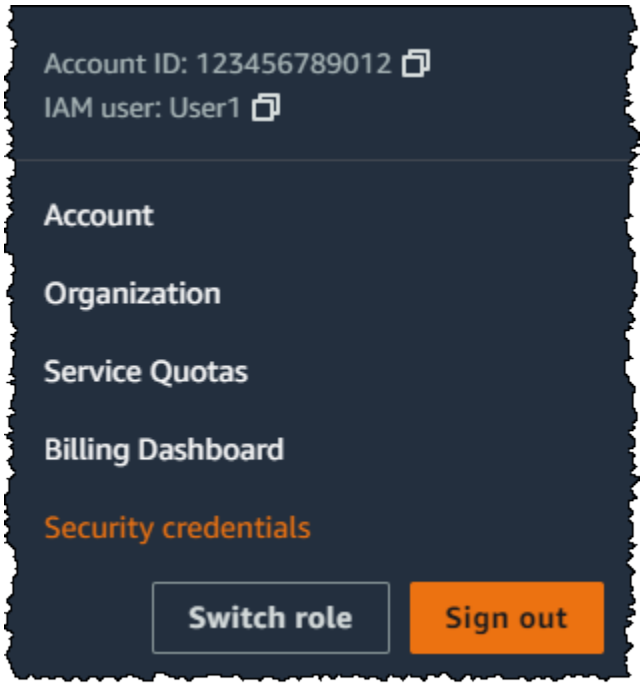
Per abilitare una passkey o una chiave di sicurezza per il tuo utente IAM (console)

1. Usa l' AWS ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM.

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella pagina dell'utente IAM selezionato, scegli la scheda Credenziali di sicurezza.
4. Nella sezione Autenticazione a più fattori (MFA), scegliere Assegna dispositivo MFA.
5. Nella pagina del nome del dispositivo MFA, inserisci un nome dispositivo, scegli Passkey o Security Key, quindi scegli Avanti.
6. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.

7. Segui le istruzioni sul tuo browser e poi scegli Continua.

Ora hai registrato la tua passkey o la chiave di sicurezza per utilizzarla con AWS. Per informazioni sull'utilizzo della tecnologia MFA con AWS Management Console, vedere. Utilizzo di dispositivi MFA con la pagina di accesso IAM

Abilita una passkey o una chiave di sicurezza per un altro utente IAM (console)

Puoi abilitare una passkey o una sicurezza per un altro utente IAM AWS Management Console solo dall'API AWS CLI o AWS , non dall'API.

Per abilitare una passkey o una sicurezza per un altro utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. In Utenti, scegli il nome dell'utente per il quale desideri abilitare l'MFA.
4. Nella pagina utente IAM selezionata, scegli la scheda Security Credentials.
5. Nella sezione Autenticazione a più fattori (MFA), scegliere Assegna dispositivo MFA.
6. Nella pagina del nome del dispositivo MFA, inserisci un nome dispositivo, scegli Passkey o Security Key, quindi scegli Avanti.
7. In Configura dispositivo, configura la tua passkey. Crea una passkey con dati biometrici come il viso o l'impronta digitale, con un pin del dispositivo oppure inserendo la chiave di sicurezza FIDO nella porta USB del computer e toccandola.
8. Segui le istruzioni sul tuo browser e poi scegli Continua.

Ora hai registrato una passkey o una chiave di sicurezza per un altro utente IAM da utilizzare. AWS Per informazioni sull'utilizzo della tecnologia MFA con AWS Management Console, vedere. Utilizzo di dispositivi MFA con la pagina di accesso IAM

Sostituire una passkey o una chiave di sicurezza

Puoi avere fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati assegnati a un utente alla volta con i tuoi Utente root dell'account AWS utenti e IAM. Se l'utente dovesse perdere l'autenticatore FIDO o in caso di sostituzione, dovrai prima disattivare il vecchio autenticatore FIDO. e quindi aggiungere un nuovo dispositivo MFA.

- Per disattivare il dispositivo correntemente associato a un utente IAM, consulta Disattivazione dei dispositivi MFA.
- Per aggiungere una nuova chiave di sicurezza FIDO per un utente IAM, consulta la sezione Abilita una passkey o una chiave di sicurezza per il tuo utente IAM (console).

Se non hai accesso a una nuova passkey o chiave di sicurezza, puoi abilitare un nuovo dispositivo MFA virtuale o un token TOTP hardware. Per istruzioni, consulta uno dei seguenti articoli:

- Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console)
- Abilitazione di un token TOTP hardware (console)

Configurazioni supportate per l'utilizzo di chiavi di accesso e chiavi di sicurezza

È possibile utilizzare le passkey FIDO2 legate al dispositivo, note anche come chiavi di sicurezza, come metodo di autenticazione a più fattori (MFA) con IAM utilizzando le configurazioni attualmente supportate. Questi includono i dispositivi FIDO2 supportati da IAM e i browser che supportano FIDO2. Prima di registrare il tuo dispositivo FIDO2, verifica di utilizzare la versione più recente del browser e del sistema operativo (OS). Le funzionalità possono comportarsi in modo diverso tra browser, autenticatori e client del sistema operativo. Se la registrazione del dispositivo non riesce su un browser, puoi provare a registrarti con un altro browser.

FIDO2 è uno standard di autenticazione aperto e un'estensione di FIDO U2F che offre lo stesso livello elevato di sicurezza basato sulla crittografia a chiave pubblica. FIDO2 è costituito dalla specifica di autenticazione Web (WebAuthn API) del W3C e dal protocollo FIDO Alliance Client-to-Authenticator Protocol (CTAP), un protocollo a livello di applicazione. CTAP consente la comunicazione tra client o piattaforma, come un browser o un sistema operativo, con un autenticatore esterno. Quando abiliti un autenticatore certificato FIDO AWS, la chiave di sicurezza crea una nuova coppia di chiavi da utilizzare solo con. AWS In primo luogo, è necessario immettere le credenziali. Quando richiesto, tocchi la chiave di sicurezza, che risponde alla sfida di autenticazione emessa da. AWS Per ulteriori informazioni sullo standard FIDO2, consulta la pagina Progetto FIDO2.

Dispositivi FIDO2 supportati da AWS

IAM supporta i dispositivi di sicurezza FIDO2 che si connettono ai tuoi dispositivi tramite USB, Bluetooth o NFC. IAM supporta anche autenticatori di piattaforme come TouchID, FaceID o Windows Hello.

> **Note**
>
> AWS richiede l'accesso alla porta USB fisica del computer per verificare il dispositivo FIDO2. Le chiavi di sicurezza non funzionano con una macchina virtuale, una connessione remota o la modalità di navigazione in incognito di un browser.

FIDO Alliance mantiene un elenco di tutti i prodotti FIDO2 compatibili con le specifiche FIDO.

Browser che supportano FIDO2

La disponibilità dei dispositivi di sicurezza FIDO2 che funzionano in un browser Web dipende dalla combinazione di browser e sistema operativo. I seguenti browser attualmente supportano l'uso delle chiavi di sicurezza:

	macOS 10.15+	Windows 10	Linux	iOS 14.5+	Android 7+
Chrome	Sì	Sì	Sì	Sì	No
Safari	Sì	No	No	Sì	No
Edge	Sì	Sì	No	Sì	No
Firefox	Sì	Sì	No	Sì	No

> **Note**
>
> La maggior parte delle versioni di Firefox che attualmente supportano FIDO2 non abilitano il supporto per impostazione predefinita. Per istruzioni su come abilitare il supporto FIDO2 in Firefox, consulta. Risoluzione dei problemi relativi alle chiavi di sicurezza FIDO

Per ulteriori informazioni sul supporto del browser per un dispositivo certificato FIDO2, ad esempio YubiKey, vedi Supporto del sistema operativo e del browser web per FIDO2 e U2F.

Plug-in del browser

AWS supporta solo i browser che supportano nativamente FIDO2. AWS non supporta l'utilizzo di plugin per aggiungere il supporto per il browser FIDO2. Alcuni plugin del browser non sono compatibili con lo standard FIDO2 e possono causare risultati imprevisti con le chiavi di sicurezza FIDO2.

Per informazioni su come disabilitare i plug-in del browser e altri suggerimenti per la risoluzione dei problemi, consulta Non riesco ad abilitare la chiave di sicurezza FIDO.

Certificazioni dei dispositivi

Acquisiamo e assegniamo le certificazioni relative ai dispositivi, come la convalida FIPS e il livello di certificazione FIDO, solo durante la registrazione di una chiave di sicurezza. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza cambia, ciò non si rifletterà automaticamente nei tag del dispositivo. Per aggiornare le informazioni di certificazione di un dispositivo, è necessario registrarlo nuovamente per recuperare le informazioni di certificazione aggiornate.

AWS fornisce i seguenti tipi di certificazione come chiavi di condizione durante la registrazione del dispositivo, ottenute da FIDO MDS: livelli di certificazione FIDO, FIPS-140-2 e FIPS-140-3. Hai la possibilità di specificare la registrazione di autenticatori specifici nelle loro policy IAM, in base al tipo e al livello di certificazione che preferisci. Per ulteriori informazioni, consulta le policy seguenti.

Policy di esempio per le certificazioni dei dispositivi

I seguenti casi d'uso mostrano policy di esempio che consentono di registrare i dispositivi MFA con certificazioni FIPS.

Argomenti

- Caso d'uso 1: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2
- Caso d'uso 2: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2 o FIDO L1
- Caso d'uso 3: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2 o FIPS-140-3 L2
- Caso d'uso 4: consenti la registrazione di dispositivi con certificazione FIPS-140-2 L2 e che supportano altri tipi di autenticazione a più fattori, come autenticatori virtuali e hardware TOTP

Caso d'uso 1: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        }
    ]
}
```

Caso d'uso 2: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2 o FIDO L1

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2",
                    "iam:FIDO-certification": "L1"
                }
            }
        }
    ]
}
```

Caso d'uso 3: consentire la registrazione di dispositivi con certificazioni FIPS-140-2 L2 o FIPS-140-3 L2

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L2"
                }
            }
        }
    ]
}
```

Caso d'uso 4: consenti la registrazione di dispositivi con certificazione FIPS-140-2 L2 e che supportano altri tipi di autenticazione a più fattori, come autenticatori virtuali e hardware TOTP

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey": "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey": "Activate",
                    "iam:FIPS-140-2-certification": "L2"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
```

```
                "Null": {
                    "iam:RegisterSecurityKey": "true"
                }
            }
        }
    ]
}
```

AWS CLI e AWS API

AWS supporta l'utilizzo di chiavi di accesso e chiavi di sicurezza solo in. AWS Management Console L'utilizzo di passkey e chiavi di sicurezza per MFA non è supportato nell'API AWS and o per AWS CLIl'accesso a operazioni API protette da MFA.

Risorse aggiuntive

- Per ulteriori informazioni sull'utilizzo delle passkey e delle chiavi di sicurezza in, vedere. AWSAbilitazione di una passkey o di una chiave di sicurezza (console)
- Per informazioni sulla risoluzione dei problemi relativi alle passkey e alle chiavi di sicurezza in AWS, vedere. Risoluzione dei problemi relativi alle chiavi di sicurezza FIDO
- Per informazioni generiche sul supporto FIDO2, consulta la pagina del Progetto FIDO2.

Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console)

Puoi utilizzare un telefono o un altro dispositivo come dispositivo di autenticazione a più fattori (MFA) virtuale. A tale scopo, installa un'app mobile conforme a RFC 6238, un algoritmo TOTP (password monouso temporanea) basato su standard. Queste app generano un codice di autenticazione a sei cifre. Poiché possono essere eseguiti su dispositivi mobili non sicuri, i dispositivi MFA virtuali potrebbero non offrire lo stesso livello di sicurezza delle chiavi di sicurezza FIDO. È consigliabile utilizzare un dispositivo MFA virtuale nell'attesa dell'approvazione di un acquisto hardware o della consegna del dispositivo hardware.

La maggior parte delle app MFA virtuali supporta la creazione di più dispositivi virtuali, consentendoti di utilizzare la stessa app per più utenti Account AWS . Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Con più dispositivi MFA, è sufficiente un solo dispositivo MFA per accedere AWS Management Console o creare una sessione tramite l' AWS CLI as quell'utente. Consigliamo di registrare più dispositivi MFA. Per le applicazioni di autenticazione, consigliamo inoltre di abilitare

le funzionalità di backup o sincronizzazione su cloud in tali app per evitare di perdere l'accesso all'account in caso di smarrimento o guasto del dispositivo che dispone delle app di autenticazione.

Per un elenco delle app MFA virtuali che è possibile utilizzare, consulta Autenticazione a più fattori. AWS richiede un'app MFA virtuale che genera un OTP a sei cifre.

Argomenti

- Autorizzazioni richieste
- Abilitazione di un dispositivo MFA virtuale per un utente IAM (console)
- Sostituzione di un dispositivo MFA virtuale

Autorizzazioni richieste

Per gestire i dispositivi MFA virtuali per l'utente IAM, è necessario disporre delle autorizzazioni dalla policy seguente: AWS: consente agli utenti IAM autenticati tramite MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza.

Abilitazione di un dispositivo MFA virtuale per un utente IAM (console)

Puoi utilizzare IAM AWS Management Console per abilitare e gestire un dispositivo MFA virtuale per un utente IAM nel tuo account. È possibile collegare tag alle risorse IAM, inclusi i dispositivi MFA virtuali, per identificare, organizzare e controllare l'accesso a tali risorse. È possibile etichettare i dispositivi MFA virtuali solo quando si utilizza l'API AWS CLI o AWS . Per abilitare e gestire un dispositivo MFA utilizzando l' AWS API AWS CLI or, vedere. Abilitazione e gestione di dispositivi MFA virtuali (AWS CLI o AWS API) Per ulteriori informazioni sul tagging delle risorse IAM, consulta Tagging delle risorse IAM.

> **Note**
>
> È necessario avere l'accesso fisico all'hardware che ospiterà il dispositivo MFA virtuale dell'utente per configurare MFA. Ad esempio, è possibile configurare MFA per un utente che utilizzerà un dispositivo MFA virtuale in esecuzione su uno smartphone. In questo caso, è necessario disporre di uno smartphone per completare la procedura guidata. Per questo motivo, è possibile consentire agli utenti di configurare e gestire i propri dispositivi MFA virtuali. In questo caso è necessario concedere agli utenti le autorizzazioni per eseguire le necessarie operazioni IAM. Per ulteriori informazioni e per un esempio di una policy IAM che concede queste autorizzazioni, consulta Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA e la policy di esempio AWS: consente agli utenti

IAM autenticati tramite MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza.

Come abilitare un dispositivo MFA virtuale per un utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Nell'elenco degli Utenti, seleziona il nome dell'utente IAM.
4. Seleziona la scheda Credenziali di sicurezza. Nella sezione Autenticazione a più fattori (MFA), scegliere Assegna dispositivo MFA.
5. Nella procedura guidata, digita un nome per il dispositivo, scegli l'app Authenticator e quindi scegli Next (Avanti).

   IAM genera e visualizza le informazioni di configurazione per il dispositivo MFA virtuale, tra cui il codice grafico QR. Il grafico è una rappresentazione della "chiave di configurazione segreta" disponibile per l'inserimento manuale sui dispositivi che non supportano i codici QR.
6. Aprire l'app MFA virtuale. Per un elenco delle app che è possibile utilizzare per ospitare i dispositivi MFA virtuali, consultare la pagina Autenticazione a più fattori.

   Se l'app MFA virtuale supporta più account o dispositivi MFA virtuali, selezionare l'opzione che consente di creare un nuovo account o dispositivo MFA virtuale.
7. Determinare se l'app MFA supporta i codici QR e procedere in uno dei seguenti modi:

   - Nella procedura guidata, scegliere Mostra codice QR ed eseguire la scansione del codice QR tramite l'app. Ad esempio, è possibile selezionare l'icona della fotocamera o un'opzione simile a Scannerizza codice ed eseguire la scansione del codice tramite la fotocamera del dispositivo.
   - Nella procedura guidata, scegli Show secret key (Mostra chiave segreta) e digitare la chiave segreta nell'app MFA.

   Al termine, il dispositivo MFA virtuale avvia la generazione di password una tantum.
8. Nella pagina Set up device (Configura il dispositivo), nella casella MFA code 1 (Codice MFA 1), digita la password monouso correntemente visualizzata nel dispositivo MFA virtuale. Attendere fino a un massimo di 30 secondi prima che il dispositivo generi una nuova password una tantum.

Quindi, digitare la seconda password monouso nella casella Codice MFA 2. Scegli Aggiungi MFA.

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, il dispositivo MFA si associa correttamente con l'utente ma il dispositivo MFA non viene sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo MFA virtuale è ora pronto per l'uso con. AWS Per informazioni sull'utilizzo della tecnologia MFA con AWS Management Console, vedere. Utilizzo di dispositivi MFA con la pagina di accesso IAM

Sostituzione di un dispositivo MFA virtuale

Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Se l'utente dovesse perdere il proprio dispositivo o in caso di sostituzione, dovrai disattivare il vecchio dispositivo e quindi aggiungere quello nuovo.

- Per disattivare il dispositivo correntemente associato a un altro utente IAM, consulta Disattivazione dei dispositivi MFA.
- Per aggiungere un dispositivo MFA virtuale sostitutivo per un altro utente IAM, segui la procedura Abilitazione di un dispositivo MFA virtuale per un utente IAM (console) descritta sopra.
- Per aggiungere un dispositivo MFA virtuale sostitutivo per Utente root dell'account AWS, segui i passaggi della procedura. Abilitazione di un dispositivo MFA virtuale per l' Utente root dell'account AWS (console)

Abilitazione di un token TOTP hardware (console)

Un dispositivo MFA hardware genera un codice numerico di sei cifre basato su un algoritmo di password monouso sincronizzato nel tempo. L'utente deve immettere un codice valido dal dispositivo quando richiesto durante la procedura di accesso. Ogni dispositivo MFA assegnato a un utente deve essere univoco; un utente non può immettere un codice dal dispositivo di un altro utente per effettuare l'autenticazione. I dispositivi MFA non possono essere condivisi tra account o utenti.

I dispositivi MFA hardware e le chiavi di sicurezza FIDO sono entrambi dispositivi fisici che si acquistano. I dispositivi hardware MFA generano codici TOTP per l'autenticazione quando accedi a. AWS Si basano sulle batterie, che potrebbero richiedere la sostituzione e la risincronizzazione nel tempo. AWS Le chiavi di sicurezza FIDO, che utilizzano la crittografia a chiave pubblica, non richiedono batterie e offrono un processo di autenticazione senza interruzioni. Consigliamo di utilizzare le chiavi di sicurezza FIDO per la loro resistenza al phishing, che forniscono un'alternativa più sicura ai dispositivi TOTP. Inoltre, le chiavi di sicurezza FIDO possono supportare più utenti IAM o root sullo stesso dispositivo, migliorandone l'utilità per la sicurezza degli account. Per specifiche e informazioni sull'acquisto per entrambi i tipi di dispositivo, consulta Multi-Factor Authentication.

È possibile abilitare un token TOTP hardware per un utente IAM dalla AWS Management Console riga di comando o dall'API IAM. Per abilitare un dispositivo MFA per il tuo Utente root dell'account AWS, consulta. Abilita un token TOTP hardware per Utente root dell'account AWS (console)

Puoi registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con i tuoi Utente root dell'account AWS utenti e IAM. Con più dispositivi MFA, è sufficiente un solo dispositivo MFA per accedere AWS Management Console o creare una sessione tramite l' AWS CLI as quell'utente.

> **Important**
>
> Ti consigliamo di abilitare più dispositivi MFA per gli utenti in modo da garantire l'accesso continuo al tuo account in caso di smarrimento del dispositivo MFA o se diventa inaccessibile.

> **Note**
>
> Per abilitare il dispositivo MFA dalla riga di comando, utilizzare `aws iam enable-mfa-device`. Per abilitare il dispositivo MFA con l'API IAM, utilizza l'operazione `EnableMFADevice`.

Argomenti

- Autorizzazioni richieste
- Abilitazione di un dispositivo MFA hardware per un utente IAM (console)
- Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)
- Sostituzione di un dispositivo MFA fisico

Autorizzazioni richieste

Per gestire un dispositivo MFA hardware per il proprio utente IAM proteggendo le operazioni sensibili correlate a MFA, è necessario disporre delle autorizzazioni dalla policy seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

Abilitazione di un dispositivo MFA hardware per un utente IAM (console)

È possibile abilitare il proprio dispositivo MFA hardware tramite AWS Management Console.

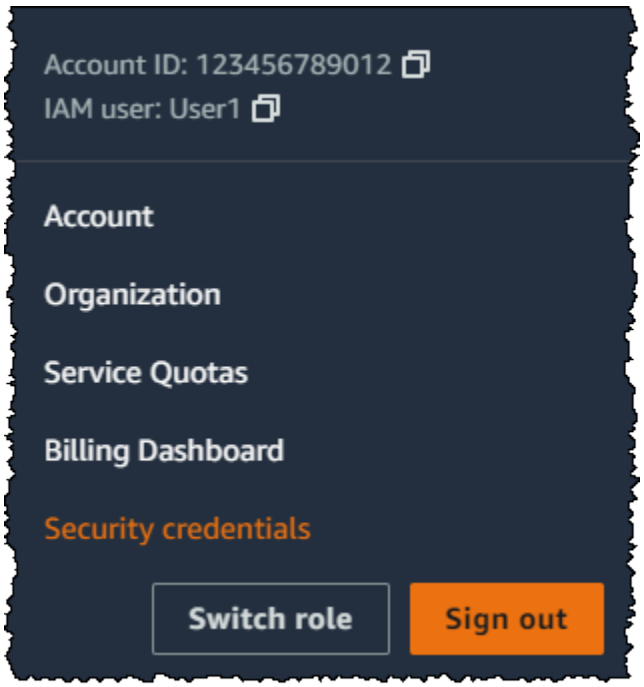
> **Note**
>
> Prima di abilitare un dispositivo MFA hardware è necessario disporre di accesso fisico al dispositivo.

Come abilitare un dispositivo MFA hardware per un utente IAM (console)

1. Utilizza l'ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. AWS

   > **Note**
   >
   > Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

   Per ottenere il tuo Account AWS ID, contatta l'amministratore.
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, nella sezione Autenticazione a più fattori, seleziona Gestione dispositivo MFA.
4. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
5. Digitare il numero di serie del dispositivo. In genere, il numero di serie è indicato sulla parte posteriore del dispositivo.
6. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.

7. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
8. Scegli Aggiungi MFA.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare

la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con AWS. Per ulteriori informazioni sull'utilizzo di MFA con la AWS Management Console, consulta Utilizzo di dispositivi MFA con la pagina di accesso IAM.

Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)

Puoi abilitare un dispositivo MFA hardware per un altro utente IAM dalla AWS Management Console.

Come abilitare un dispositivo MFA hardware per un altro utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome del segreto per il quale desideri abilitare la rotazione.
4. Seleziona la scheda Credenziali di sicurezza. Nella sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori), scegliere Assign MFA device (Assegna dispositivo MFA).
5. Nella procedura guidata, digitate il nome del dispositivo, scegliete il token Hardware TOTP e quindi scegliete Avanti.
6. Digitare il numero di serie del dispositivo. In genere, il numero di serie è indicato sulla parte posteriore del dispositivo.
7. Nella casella MFA code 1 (Codice MFA 1) digitare il numero di sei cifre visualizzato nel dispositivo MFA. Per visualizzare il numero, potrebbe essere necessario premere il pulsante sul lato anteriore del dispositivo.

8. Attendere 30 secondi per consentire al dispositivo di aggiornare il codice, quindi digitare il nuovo numero a sei cifre nella casella MFA code 2 (Codice MFA 2). Per visualizzare il secondo numero, potrebbe essere necessario premere nuovamente il pulsante sul lato anteriore del dispositivo.
9. Scegli Aggiungi MFA.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

Il dispositivo è pronto per l'uso con AWS. Per ulteriori informazioni sull'utilizzo di MFA con la AWS Management Console, consulta Utilizzo di dispositivi MFA con la pagina di accesso IAM.

Sostituzione di un dispositivo MFA fisico

Puoi avere fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati assegnati a un utente alla volta con i tuoi Utente root dell'account AWS utenti e IAM. Se l'utente dovesse perdere il proprio dispositivo o in caso di sostituzione, dovrai disattivare il vecchio dispositivo e quindi aggiungere quello nuovo.

- Per disattivare il dispositivo associato al momento con un utente, consultare Disattivazione dei dispositivi MFA.
- Per aggiungere un dispositivo MFA hardware sostitutivo per un utente IAM, segui la procedura Abilitazione di un dispositivo MFA hardware per un altro utente IAM (console)descritta prima in questo argomento.
- Per aggiungere un token TOTP hardware sostitutivo per Utente root dell'account AWS, segui i passaggi della procedura descritta in Abilita un token TOTP hardware per Utente root dell'account AWS (console) precedenza in questo argomento.

Abilitazione e gestione di dispositivi MFA virtuali (AWS CLI o AWS API)

Puoi utilizzare AWS CLI comandi o operazioni AWS API per abilitare un dispositivo MFA virtuale per un utente IAM. Non è possibile abilitare un dispositivo MFA per il Utente root dell'account AWS AWS CLI, AWS API, Tools for Windows PowerShell o qualsiasi altro strumento da riga di comando. Tuttavia, è possibile utilizzare il AWS Management Console per abilitare un dispositivo MFA per l'utente root.

Quando abiliti un dispositivo MFA da AWS Management Console, la console esegue più passaggi per te. Se invece crei AWS CLI un dispositivo virtuale utilizzando Tools for Windows PowerShell o AWS API, devi eseguire i passaggi manualmente e nell'ordine corretto. Ad esempio, per creare un dispositivo MFA virtuale, è necessario creare l'oggetto IAM ed estrarre il codice sotto forma di stringa o di codice grafico QR. Quindi è necessario sincronizzare il dispositivo e associarlo a un utente IAM. Consulta la sezione Esempi di New-IAMVirtualMFADevice per ulteriori dettagli. Per un dispositivo fisico, si salta la fase di creazione per andare direttamente a sincronizzare il dispositivo e associarlo con l'utente.

È possibile collegare tag alle risorse IAM, inclusi i dispositivi MFA virtuali, per identificare, organizzare e controllare l'accesso a tali risorse. È possibile etichettare i dispositivi MFA virtuali solo quando si utilizza l'API AWS CLI o AWS .

Un utente IAM che utilizza l'SDK o la CLI può abilitare un dispositivo MFA aggiuntivo chiamando `EnableMFADevice` o disattivarlo chiamando `DeactivateMFADevice`. Per eseguire correttamente questa operazione, devono prima chiamare `GetSessionToken` e inviare i codici MFA con un dispositivo MFA esistente. Questa chiamata restituisce credenziali di sicurezza temporanee che possono quindi essere utilizzate per firmare operazioni API che richiedono l'autenticazione MFA. Per un esempio di richiesta e risposta, consulta `GetSessionToken`: credenziali temporanee per gli utenti in ambienti non attendibili.

Creazione dell'entità del dispositivo virtuale in IAM in modo che rappresenti un dispositivo MFA virtuale

Questi comandi forniscono un ARN per il dispositivo che viene usato al posto di un numero di serie in molti dei seguenti comandi.

- AWS CLI: `aws iam create-virtual-mfa-device`
- AWS API: `CreateVirtualMFADevice`

Per abilitare un dispositivo MFA da utilizzare con AWS

Questi comandi sincronizzano il dispositivo AWS e lo associano a un utente. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie.

> **Important**
>
> La richiesta deve essere inviata immediatamente dopo la generazione dei codici di autenticazione. Se dopo avere generato i codici attendi troppo a lungo prima di inviare

la richiesta, il dispositivo MFA si associerà correttamente con l'utente, ma perderà la sincronizzazione. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. In questo caso, è possibile risincronizzare il dispositivo utilizzando i comandi descritti di seguito.

- AWS CLI: `aws iam enable-mfa-device`
- AWS API: `EnableMFADevice`

Per disattivare un dispositivo

Utilizzare questi comandi per dissociare il dispositivo dall'utente e disattivarlo. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie. È inoltre necessario eliminare separatamente l'entità del dispositivo virtuale.

- AWS CLI: `aws iam deactivate-mfa-device`
- AWS API: `DeactivateMFADevice`

Elenco delle entità del dispositivo MFA virtuale

Utilizzare questi comandi per elencare le entità di un dispositivo MFA virtuale.

- AWS CLI: `aws iam list-virtual-mfa-devices`
- AWS API: `ListVirtualMFADevices`

Per applicare tag a un dispositivo MFA virtuale

Utilizzare questi comandi per applicare tag a un dispositivo MFA virtuale.

- AWS CLI: `aws iam tag-mfa-device`
- AWS API: `TagMFADevice`

Per elencare i tag per un dispositivo MFA virtuale

Utilizzare questi comandi per elencare i tag collegati a un dispositivo MFA virtuale.

- AWS CLI: `aws iam list-mfa-device-tags`

- AWS API: `ListMFADeviceTags`

Per rimuovere i tag da un dispositivo MFA virtuale

Utilizzare questi comandi per rimuovere i tag collegati a un dispositivo MFA virtuale.

- AWS CLI: `aws iam untag-mfa-device`
- AWS API: `UntagMFADevice`

Risincronizzare un dispositivo MFA

Usa questi comandi se il dispositivo genera codici che non sono accettati da AWS. Se il dispositivo è virtuale, utilizzare l'ARN di un dispositivo virtuale come numero di serie.

- AWS CLI: `aws iam resync-mfa-device`
- AWS API: `ResyncMFADevice`

Come eliminare un'entità del dispositivo MFA virtuale in IAM

Dopo che il dispositivo è stato dissociato da parte dell'utente, è possibile eliminare l'entità del dispositivo.

- AWS CLI: `aws iam delete-virtual-mfa-device`
- AWS API: `DeleteVirtualMFADevice`

Per recuperare un dispositivo MFA virtuale che è stato smarrito o non funziona

Potrebbe accadere che il dispositivo di un utente che ospita l'app MFA virtuale venga smarrito o sostituito o non funzioni. Quando ciò si verifica, l'utente non può recuperarlo autonomamente. L'utente deve contattare un amministratore per disattivare il dispositivo. Per ulteriori informazioni, consulta Cosa fare se un dispositivo MFA viene smarrito o smette di funzionare?.

Verifica dello stato MFA

Utilizza la console IAM per verificare se un utente Utente root dell'account AWS o un utente IAM ha abilitato un dispositivo MFA valido.

Come verificare lo stato MFA di un utente root

1. Accedi a AWS Management Console con le tue credenziali utente root, quindi apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).
3. Controlla in Multi-Factor Authentication (MFA) per verificare se l'autenticazione MFA è abilitata o disabilitata. Se l'autenticazione MFA non è abilitata, viene visualizzato un simbolo di avviso ().

Per abilitare l'autenticazione MFA per l'account, consultare uno dei seguenti articoli:

- Abilitazione di un dispositivo MFA virtuale per l' Utente root dell'account AWS (console)
- Abilita una passkey o una chiave di sicurezza per Utente root dell'account AWS (console)
- Abilita un token TOTP hardware per Utente root dell'account AWS (console)

Come verificare lo stato MFA degli utenti IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna MFA alla tabella degli utenti mediante la procedura seguente:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ().
   b. In Manage Columns (Gestisci colonne) selezionare MFA.
   c. (Opzionale) Deselezionare la casella di controllo per le intestazioni di colonna che non si desidera visualizzare nella tabella utenti.
   d. Selezionare Close (Chiudi) per tornare all'elenco degli utenti.
4. La colonna MFA fornisce informazioni sul dispositivo MFA abilitato. Se per l'utente non è attivo alcun dispositivo MFA, la console visualizza None (Nessuno). Se l'utente dispone di un dispositivo MFA abilitato, la colonna MFA mostra il tipo di dispositivo abilitato con il valore Virtual (Virtuale), Security Key (Chiave di sicurezza), Hardware o SMS.

> **Note**
>
> AWS è terminato il supporto per l'abilitazione dell'autenticazione a più fattori (MFA) tramite SMS. Consigliamo ai clienti con utenti IAM che utilizzano la MFA basata su messaggi di testo SMS di passare a uno dei seguenti metodi alternativi di autenticazione a più fattori: dispositivo MFA virtuale (basato su software), chiave di sicurezza FIDO o dispositivo MFA basato su hardware. Puoi identificare gli utenti nel tuo account con un dispositivo MFA SMS assegnato. Per farlo, vai alla console IAM, scegli Users (Utenti) dal riquadro di navigazione e individua gli utenti con SMS nella colonna della tabella MFA.

5. Per visualizzare ulteriori informazioni sul dispositivo MFA per un utente, selezionare il nome dell'utente di cui si desidera verificare lo stato MFA. Quindi, selezionare la scheda Security credentials (Credenziali di sicurezza).
6. Se per l'utente non è attivo alcun dispositivo MFA, la console visualizza No MFA devices (Nessun dispositivo MFA). Assegna un dispositivo MFA per migliorare la sicurezza del AWS tuo ambiente nella sezione Autenticazione a più fattori (MFA). Se l'utente ha i dispositivi MFA abilitati, la sezione Autenticazione a più fattori (MFA) mostra i dettagli dei dispositivi:

   - Il nome del dispositivo
   - Il tipo di dispositivo
   - L'identificatore del dispositivo, ad esempio un numero di serie per un dispositivo fisico o l'ingresso ARN AWS per un dispositivo virtuale
   - Quando il dispositivo è stato creato

Per rimuovere o risincronizzare un dispositivo, scegli il pulsante d'opzione accanto al dispositivo e scegli Remove (Rimuovi) o Resync (Risincronizza).

Per ulteriori informazioni sull'abilitazione di MFA, consultare quanto segue:

- Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console)
- Abilitazione di una passkey o di una chiave di sicurezza (console)
- Abilitazione di un token TOTP hardware (console)

Risincronizzazione dei dispositivi MFA virtuali e hardware

È possibile utilizzarli AWS per risincronizzare i dispositivi di autenticazione a più fattori (MFA) virtuali e hardware. Se il dispositivo dell'utente non è sincronizzato quando si tenta di utilizzarlo, il tentativo di accesso dell'utente non riesce e IAM richiede all'utente di risincronizzare il dispositivo.

> **Note**
>
> Le chiavi di sicurezza FIDO sono sempre sincronizzate. Se una chiave di sicurezza FIDO viene smarrita o danneggiata, puoi disattivarla. Per istruzioni sulla disattivazione dei dispositivi MFA, consulta Come disattivare un dispositivo MFA per un altro utente IAM (console).

In qualità di AWS amministratore, puoi risincronizzare i dispositivi MFA virtuali e hardware degli utenti IAM se non vengono sincronizzati.

Se il tuo dispositivo Utente root dell'account AWS MFA non funziona, puoi risincronizzarlo utilizzando la console IAM con o senza completare la procedura di accesso. Se non riesci a risincronizzare correttamente il dispositivo, potrebbe essere necessario dissociarlo e associarlo nuovamente. Per ulteriori informazioni su come effettuare tale operazione, consulta Disattivazione dei dispositivi MFA e Abilitazione dei dispositivi MFA per gli utenti in AWS.

Argomenti

- Autorizzazioni richieste
- Risincronizzazione dei dispositivi MFA virtuali e hardware (console IAM)
- Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS CLI)
- Risincronizzazione di dispositivi MFA (API) virtuali e hardware AWS

Autorizzazioni richieste

Per sincronizzare nuovamente i dispositivi MFA virtuali o hardware per l'utente IAM, è necessario disporre delle autorizzazioni dalla politica seguente. Questa politica non consente di creare o disattivare un dispositivo.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "AllowListActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToViewAndManageTheirOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "BlockAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

Risincronizzazione dei dispositivi MFA virtuali e hardware (console IAM)

Puoi utilizzare la console IAM per risincronizzare i dispositivi MFA virtuali e hardware.

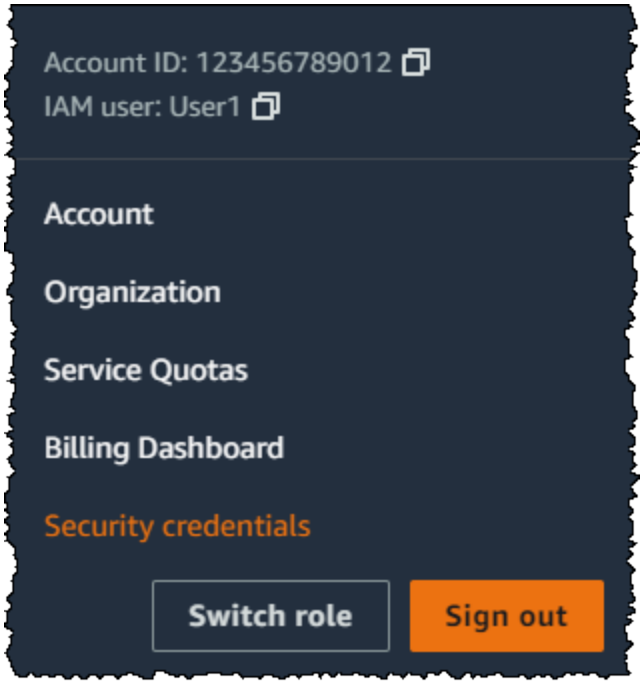
Come risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (console)

1. Utilizza l'ID o l'alias dell'account, il nome utente IAM e la password per accedere alla console IAM. AWS

> **Note**
>
> Per comodità, la pagina di AWS accesso utilizza un cookie del browser per ricordare il nome utente IAM e le informazioni sull'account. Se in precedenza è stato eseguito l'accesso con un utente diverso, scegli il link Accedi a un account differente nella parte inferiore della pagina per ritornare alla pagina principale di accesso. Da lì, puoi digitare l'ID o l'alias dell'account per essere reindirizzato alla pagina di accesso utente IAM relativa al tuo AWS account.

Per ottenere il tuo Account AWS ID, contatta l'amministratore.

2. Seleziona il nome utente in alto a destra nella barra di navigazione e scegli Security credentials (Credenziali di sicurezza).



3. Nella scheda Credenziali AWS IAM, nella sezione Autenticazione a più fattori (MFA), scegli il pulsante d'opzione accanto al dispositivo MFA e seleziona Risincronizza.
4. Digitare i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, la richiesta sembra riuscita ma il dispositivo non è comunque sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Come risincronizzare un dispositivo MFA virtuale o hardware per un altro utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, selezionare Users (Utenti)e il nome dell'utente con dispositivo MFA da risincronizzare.
3. Selezionare la scheda Security Credentials (Credenziali di sicurezza). Nella sezione Multi-factor authentication (MFA) (Autenticazione a più fattori (MFA), scegli il pulsante d'opzione accanto al dispositivo MFA e scegli Resync (Risincronizza).
4. Digitare i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo a lungo per inviare la richiesta, la richiesta sembra riuscita ma il dispositivo non è comunque sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Come risincronizzare l'autenticazione MFA dell'utente root prima dell'accesso (console)

1. Nella pagina Accesso ad Amazon Web Services con dispositivo di autenticazione, seleziona Problemi con il tuo dispositivo di autenticazione? Fai clic qui.

> **Note**
>
> È possibile che il testo visualizzato sia differente, ad esempio Sign in using MFA (Accesso con un dispositivo MFA) e Troubleshoot your authentication device

(Risoluzione dei problemi del dispositivo di autenticazione). Tuttavia, le funzionalità sono identiche.

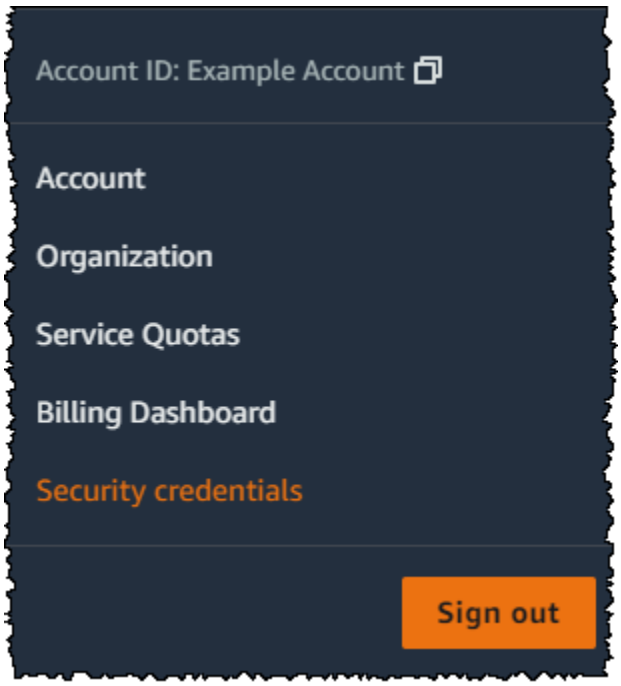
2. Nella sezione Re-Sync With Our Servers (Risincronizzazione con i nostri server), digitare i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi selezionare Re-sync authentication device (Risincronizza dispositivo di autenticazione).
3. Se necessario, digitare di nuovo la password e selezionare Accedi. Quindi completare l'accesso utilizzando il dispositivo MFA.

Come risincronizzare l'autenticazione MFA dell'utente root dopo l'accesso (console)

1. Accedi alla console IAM come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

> **Note**
>
> Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Sul lato destro della barra di navigazione, seleziona il nome dell'account, quindi Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Espandere la sezione Multi-Factor Authentication (MFA) (Autenticazione a più fattori, MFA) della pagina.
4. Seleziona il pulsante d'opzione accanto al dispositivo e scegli Resync (Risincronizza).
5. Nella finestra di dialogo Resync MFA device (Risincronizza dispositivo MFA), digita i successivi due codici generati in sequenza dal dispositivo in MFA code 1 (Codice MFA 1) e MFA code 2 (Codice MFA 2). Quindi scegli Resync (Risincronizza).

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si attende troppo a lungo per inviare la richiesta dopo la generazione dei codici, il dispositivo MFA verrà associato all'utente, ma non verrà sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo.

Risincronizzazione dei dispositivi MFA virtuali e hardware (AWS CLI)

Puoi risincronizzare i dispositivi MFA virtuali e hardware da AWS CLI.

Come risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (AWS CLI)

Al prompt dei comandi, esegui il resync-mfa-device comando aws iam:

- Dispositivo MFA virtuale: specificare l'Amazon Resource Name (ARN) del dispositivo come numero di serie.

```
aws iam resync-mfa-device --user-name Richard --serial-number
 arn:aws:iam::123456789012:mfa/RichardsMFA --authentication-code1 123456 --
authentication-code2 987654
```

- Dispositivo MFA hardware: specificare il numero di serie del dispositivo hardware come numero di serie. Il formato è specifico del fornitore. Ad esempio, puoi acquistare un token gemalto da Amazon. Il suo numero di serie è in genere quattro lettere seguite da quattro numeri.

```
aws iam resync-mfa-device --user-name Richard --serial-number ABCD12345678 --
authentication-code1 123456 --authentication-code2 987654
```

> **Important**
>
> Inviare la richiesta immediatamente dopo la generazione dei codici. Se si generano i codici e si attende troppo lungo per inviare la richiesta, la richiesta ha esito negativo perché i codici scadono dopo un breve periodo di tempo.

Risincronizzazione di dispositivi MFA (API) virtuali e hardware AWS

In IAM è disponibile una chiamata API che esegue la sincronizzazione. In questo caso, ti consigliamo di fornire agli utenti del dispositivo MFA virtuale e hardware l'autorizzazione per accedere a questa chiamata API. Quindi, crea uno strumento basato su tale chiamata API per consentire agli utenti di risincronizzare i propri dispositivi ogni volta che è necessario.

Per risincronizzare un dispositivo MFA virtuale o hardware per un utente IAM (API)AWS

- Invia la richiesta ResyncMFADevice.

Disattivazione dei dispositivi MFA

In caso di problemi di accesso con un dispositivo con autenticazione a più fattori (MFA) come utente IAM, contatta il tuo amministratore per assistenza.

In qualità di amministratore, puoi disattivare il dispositivo per un altro utente IAM. In questo modo, l''utente potrà effettuare l'accesso senza utilizzare MFA. Questa soluzione può essere temporanea,

se il dispositivo MFA è momentaneamente non disponibile o in attesa che venga sostituito. Tuttavia, ti consigliamo di abilitare un nuovo dispositivo per l'utente quanto prima possibile. Per informazioni su come abilitare un nuovo dispositivo MFA, consultare the section called “Abilitazione dei dispositivi MFA”.

> **Note**
>
> Se utilizzi l'API o desideri AWS CLI eliminare un utente dal tuo Account AWS, devi disattivare o eliminare il dispositivo MFA dell'utente. Tale modifica fa paarte del processo di rimozione dell'utente. Per ulteriori informazioni sull'eliminazione degli utenti, consultare Gestione degli utenti IAM.

Argomenti

Disattivazione dei dispositivi MFA (console)

Come disattivare un dispositivo MFA per un altro utente IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Per disattivare un dispositivo MFA, scegli il nome dell'utente a cui appartiene l'MFA da rimuovere.
4. Selezionare la scheda Security Credentials (Credenziali di sicurezza).
5. In Multi-factor authentication (MFA) (Autenticazione a più fattori (MFA), scegli il pulsante d'opzione accanto al dispositivo MFA, quindi scegli Remove (Rimuovi) e poi di nuovo Remove (Rimuovi).

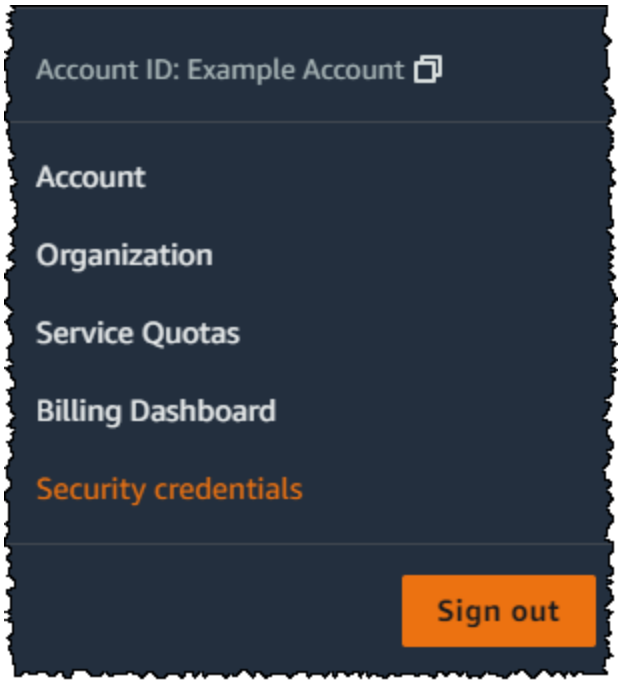
   Il dispositivo viene rimosso da AWS. Non può essere utilizzato per accedere o autenticare le richieste finché non viene riattivato e associato a un AWS utente o. Utente root dell'account AWS

Per disattivare il dispositivo MFA per un Utente root dell'account AWS (console)

1. Accedi alla console IAM come proprietario dell'account selezionando Utente root e inserendo il tuo Account AWS indirizzo email. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > Come utente root, non puoi accedere alla pagina Sign in as IAM user (Accedi come utente IAM). Se viene visualizzata la pagina Sign in as IAM user (Accedi come utente IAM), scegli Sign in using root user email (Accedi con l'indirizzo e-mail dell'utente root) nella parte inferiore della pagina. Per informazioni sull'accesso come utente root, consulta Accedere AWS Management Console come utente root nella Guida per l'Accedi ad AWS utente.

2. Sul lato destro della barra di navigazione, seleziona il nome dell'account, quindi Security Credentials (Credenziali di sicurezza). Se necessario, seleziona Continue to Security Credentials (Continua con le credenziali di sicurezza).



3. Nella sezione Multi-factor authentication (MFA) (Autenticazione a più fattori (MFA), scegli il pulsante d'opzione accanto al dispositivo MFA da disattivare e scegli Remove (Rimuovi).
4. Scegli Rimuovi.

Il dispositivo MFA è disattivato per l' Account AWS. Controlla l'e-mail associata al tuo indirizzo Account AWS per ricevere un messaggio di conferma da Amazon Web Services. L'e-mail ti informa che la tua autenticazione multifattore (MFA) di Amazon Web Services è stata disattivata. Il messaggio verrà inviato da `@amazon.com` o `@aws.amazon.com`.

Disattivazione dei dispositivi MFA (AWS CLI)

Come disattivare un dispositivo MFA per un utente IAM (AWS CLI)

- Eseguire il comando: `aws iam deactivate-mfa-device`

Disattivazione dei dispositivi MFA (API)AWS

Per disattivare un dispositivo MFA per un utente AWS IAM (API)

- Richiamare l'operazione: `DeactivateMFADevice`

Cosa fare se un dispositivo MFA viene smarrito o smette di funzionare?

Se il dispositivo MFA virtuale o il token TOTP hardware sembra funzionare correttamente, ma non è possibile utilizzarlo per accedere alle AWS risorse, potrebbe non essere sincronizzato con. AWS Per informazioni sulla sincronizzazione di un dispositivo MFA virtuale o hardware, consulta Risincronizzazione dei dispositivi MFA virtuali e hardware. Le chiavi di sicurezza FIDO sono sempre sincronizzate.

Se il dispositivo di Utente root dell'account AWS autenticazione a più fattori (MFA) viene smarrito, danneggiato o non funziona, puoi ripristinare l'accesso al tuo account. Gli utenti IAM devono contattare un amministratore per disattivare il dispositivo.

> **Important**
>
> Ti consigliamo di abilitare più dispositivi MFA per gli utenti IAM per garantire l'accesso continuo al tuo account in caso di smarrimento del dispositivo MFA o se diventa inaccessibile. È possibile registrare fino a otto dispositivi MFA di qualsiasi combinazione dei tipi di MFA attualmente supportati con l'utente root Account AWS e gli utenti IAM.

Ripristino di un dispositivo MFA per l'utente root

Se il dispositivo di Utente root dell'account AWS autenticazione a più fattori (MFA) viene smarrito, danneggiato o non funziona, puoi accedere utilizzando un altro dispositivo MFA registrato sullo stesso. Utente root dell'account AWS Se l'utente root ha un solo dispositivo MFA abilitato, potrai utilizzare metodi di autenticazione alternativi. Ciò significa che se non sei in grado di effettuare l'accesso tramite il dispositivo MFA, puoi farlo tramite la verifica dell'identità, utilizzando l'e-mail e il numero di telefono di contatto principale registrati nell'account.

Prima di utilizzare i metodi di autenticazione alternativi per accedere come utente root, assicurati di avere accesso all'e-mail e al numero di telefono di contatto principale associati all'account. Se devi aggiornare il numero di telefono di contatto principale, puoi accedere come utente IAM con accesso da Administrator (Amministratore) anziché da utente root. Per ulteriori istruzioni sull'aggiornamento delle informazioni di contatto dell'account, consulta Modifica le informazioni di contatto nella Guida per l'utente AWS Billing . Se non hai accesso a un'e-mail e al numero di telefono di contatto principale, contatta AWS Support.

> **Important**
>
> Ti consigliamo di mantenere aggiornati l'indirizzo email e il numero di telefono di contatto collegati all'utente root per ripristinare correttamente l'account. Per ulteriori informazioni, consulta Aggiornare il contatto principale per Account AWS nella Guida di riferimento AWS Account Management .

Per accedere utilizzando fattori di autenticazione alternativi come Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.
2. Nella pagina Verifica aggiuntiva richiesta, seleziona un metodo MFA con cui eseguire l'autenticazione e scegli Avanti.

> **Note**
>
> È possibile che venga visualizzato un testo alternativo, ad esempio Sign in using MFA (Accedi utilizzando la MFA), Troubleshoot your authentication device (Risolvi i problemi del tuo dispositivo di autenticazione) oppure Troubleshoot MFA (Risolvi i problemi di MFA), ma la funzionalità è la stessa. Se non riesci a utilizzare i fattori alternativi

di autenticazione per verificare l'indirizzo e-mail e il numero di telefono di contatto principale, contatta AWS Support per disattivare il dispositivo MFA.

3. A seconda del tipo di MFA in uso, verrà visualizzata una pagina diversa, ma l'opzione Risoluzione dei problemi relativi all'autenticazione MFA funziona comunque. Nella pagina Verifica aggiuntiva richiesta o nella pagina Autenticazione a più fattori, scegli Risoluzione dei problemi relativi all'autenticazione MFA.
4. Se necessario, digitare di nuovo la password e selezionare Accedi.
5. Nella pagina Risoluzione dei problemi del dispositivo di autenticazione, in Accedi utilizzando fattori alternativi di autenticazione, scegli Accedi utilizzando fattori alternativi.
6. Nella pagina Accedi utilizzando fattori di autenticazione alternativi, autentica il tuo account verificando l'indirizzo e-mail e scegli Invia email di verifica.
7. Controlla l'e-mail associata alla tua Account AWS per un messaggio proveniente da Amazon Web Services (recover-mfa-no-reply@verify .signin.aws). Seguire le istruzioni nel messaggio.

   Se il messaggio non fosse presente nell'account e-mail, controllare la cartella spam o tornare al browser e selezionare Resend the email (Rinvia l'e-mail).
8. Una volta verificato l'indirizzo e-mail, puoi continuare con l'autenticazione dell'account. Per verificare il numero di telefono di contatto principale, scegli Call me now (Chiamami ora).
9. Rispondi alla chiamata da AWS e, quando richiesto, inserisci il numero a 6 cifre del sito Web sulla tastiera del telefono. AWS

   Se non ricevi una chiamata da AWS, scegli Accedi per accedere nuovamente alla console e ricominciare da capo. In alternativa, consulta la sezione Dispositivo di autenticazione a più fattori (MFA) perso o inutilizzabile per contattare il supporto e richiedere assistenza.
10. Una volta effettuata la verifica del numero di telefono, è possibile effettuare l'accesso all'account selezionando Sign in to the console (Accedi alla console).
11. La prossima fase varia a seconda del tipo di MFA in uso:

    - Se si utilizza un dispositivo MFA virtuale, rimuovere l'account dal dispositivo. Quindi passa alla pagina Credenziali di sicurezza AWS ed elimina l'entità del vecchio dispositivo MFA virtuale prima di crearne una nuova.
    - Se utilizzi una chiave di sicurezza FIDO, visita la pagina Credenziali di sicurezza AWS e disattiva la chiave di sicurezza FIDO precedente prima di abilitarne una nuova.
    - Per un token TOTP hardware, contatta il provider di terza parte per assistenza con la riparazione o la sostituzione del dispositivo. L'accesso tramite fattori alternativi di

autenticazione può essere utilizzato fino alla ricezione di un nuovo dispositivo. Una volta ottenuto il nuovo dispositivo MFA hardware, passa alla pagina Credenziali di sicurezza AWS ed elimina l'entità del dispositivo MFA hardware precedente prima di crearne una nuova.

> **Note**
>
> Non è necessario sostituire il dispositivo MFA smarrito o rubato con lo stesso tipo di dispositivo. Ad esempio, se la chiave di sicurezza FIDO viene danneggiata e ne ordini una nuova, potrai utilizzare il dispositivo MFA virtuale o il token TOTP hardware finché non ricevi a nuova chiave di sicurezza FIDO.

> **Important**
>
> In caso di smarrimento o furto del dispositivo MFA, dopo aver effettuato l'accesso utilizzando fattori di autenticazione alternativi e aver installato il dispositivo MFA sostitutivo, modifica la password dell'utente root nel caso in cui un utente malintenzionato abbia rubato il dispositivo di autenticazione e possa avere la tua password attuale. Per ulteriori informazioni, consulta Modifica della password per l' Utente root dell'account AWS nella Guida di riferimento di AWS Account Management .

Ripristino di un dispositivo MFA dell'utente IAM

Se sei un utente IAM e il dispositivo viene perso o smette di funzionare, non puoi recuperarlo da solo, ma devi contattare un amministratore per disattivare il dispositivo. Quindi puoi abilitare un nuovo dispositivo.

Come ottenere assistenza per un dispositivo MFA in qualità di utente IAM

1. Contatta l' AWS amministratore o l'altra persona che ti ha fornito il nome utente e la password per l'utente IAM. L'amministratore deve disattivare il dispositivo MFA, come descritto in Disattivazione dei dispositivi MFA, in modo da consentire l'accesso.
2. La prossima fase varia a seconda del tipo di MFA in uso:

   - Se si utilizza un dispositivo MFA virtuale, rimuovere l'account dal dispositivo. Attivare il dispositivo virtuale come descritto in Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale (Console).

- Se utilizzi una chiave di sicurezza FIDO, contatta il fornitore di terza parte per ricevere assistenza con la sostituzione del dispositivo. Quando ricevi la nuova chiave di sicurezza FIDO, devi abilitarla come descritto nella sezione Abilitazione di una passkey o di una chiave di sicurezza (console).
- Per un token TOTP hardware, contatta il provider di terza parte per assistenza con la riparazione o la sostituzione del dispositivo. Una volta ottenuto il nuovo dispositivo MFA fisico, abilitarlo nel modo descritto in Abilitazione di un token TOTP hardware (console).

> **Note**
>
> Non è necessario sostituire il dispositivo MFA smarrito o rubato con lo stesso tipo di dispositivo. Puoi avere fino a otto dispositivi MFA in una qualsiasi combinazione. Ad esempio, se la chiave di sicurezza FIDO viene danneggiata e ne ordini una nuova, potrai utilizzare il dispositivo MFA virtuale o il token TOTP hardware finché non ricevi a nuova chiave di sicurezza FIDO.

3. Se il dispositivo MFA è stato smarrito o rubato, modificare anche la password nel caso in cui chi si è impossessato del dispositivo di autenticazione possieda anche la password attualmente in uso. Per ulteriori informazioni, consulta Gestione delle password per gli utenti IAM

Configurazione dell'accesso alle API protetto da MFA

Con le policy IAM, è possibile specificare le operazioni API che un utente è autorizzato a chiamare. In alcuni casi, è consigliabile implementare un livello di sicurezza aggiuntivo, richiedendo agli utenti di eseguire la multi-factor authentication (MFA) in AWS prima di eseguire operazioni particolarmente sensibili.

Potrebbe ad esempio esserci una policy che permette a un utente di eseguire le operazioni `RunInstances`, `DescribeInstances` e `StopInstances` di Amazon EC2. Ma potresti voler limitare un'azione distruttiva come `TerminateInstances` e assicurarti che gli utenti possano eseguire tale azione solo se si autenticano con un dispositivo AWS MFA.

Argomenti

- Scenario: Protezione MFA per le risorse che hanno policy basate su risorse

Panoramica

L'aggiunta della protezione MFA alle operazioni API prevede le operazioni seguenti:

1. L'amministratore configura un dispositivo AWS MFA per ogni utente che deve effettuare richieste API che richiedono l'autenticazione MFA. Questo processo viene descritto in Abilitazione dei dispositivi MFA per gli utenti in AWS.
2. L'amministratore crea politiche per gli utenti che includono un `Condition` elemento che verifica se l'utente si è autenticato con un dispositivo AWS MFA.
3. L'utente richiama una delle operazioni AWS STS API che supportano i parametri MFA AssumeRoleo GetSessionToken, a seconda dello scenario per la protezione MFA, come spiegato più avanti. Durante la chiamata, l'utente include l'ID dispositivo per il dispositivo associato all'utente. L'utente include anche la password una tantum a tempo (TOTP) generata dal dispositivo. In entrambi i casi, l'utente ottiene le credenziali di sicurezza temporanee che può quindi usare per effettuare richieste aggiuntive in AWS.

> **Note**
>
> La protezione MFA per le operazioni API di un servizio è disponibile solo se il servizio supporta le credenziali di sicurezza temporanee. Per un elenco di questi servizi, consulta Utilizzo di credenziali di sicurezza temporanee per accedere ad AWS.

Se l'autorizzazione AWS fallisce, restituisce un messaggio di errore di accesso negato (come accade per qualsiasi accesso non autorizzato). Con le politiche API protette da MFA, AWS nega l'accesso alle operazioni API specificate nelle politiche se l'utente tenta di richiamare un'operazione API senza un'autenticazione MFA valida. L'operazione viene rifiutata anche se il timestamp della richiesta di operazione API è al di fuori dell'intervallo consentito specificato nella policy. L'utente deve essere autenticato di nuovo con MFA richiedendo nuove credenziali di sicurezza temporanee con un codice MFA e il numero di serie del dispositivo.

Policy IAM con condizioni MFA

Le policy con condizioni MFA possono essere collegate a:

- Un utente o un gruppo IAM

- Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
- La policy di attendibilità di un ruolo IAM che può essere assunto da un utente

Puoi usare una condizione MFA in una policy per controllare le proprietà seguenti:

- Esistenza: per verificare semplicemente che l'utente abbia eseguito l'autenticazione con MFA, controlla che la chiave `aws:MultiFactorAuthPresent` sia `True` in una condizione `Bool`. La chiave è presente solo quando l'utente esegue l'autenticazione con credenziali a breve termine. Le credenziali a lungo termine, ad esempio le chiavi di accesso, non includono questa chiave.
- Durata: se desideri concedere l'accesso solo per un periodo di tempo specificato dopo l'autenticazione MFA, usa una condizione di tipo numerico per confrontare la validità della chiave `aws:MultiFactorAuthAge` con un valore (ad esempio 3.600 secondi). Ricordati che la chiave `aws:MultiFactorAuthAge` non è presente se non è stata usata l'autenticazione MFA.

L'esempio seguente mostra la policy di attendibilità di un ruolo IAM che include una condizione MFA da testare per verificare l'esistenza dell'autenticazione MFA. Con questa politica, gli utenti Account AWS specificati nell'`Principal`elemento (sostituendo `ACCOUNT-B-ID` con un Account AWS ID valido) possono assumere il ruolo a cui è associata questa politica. ma solo se si sono autenticati tramite MFA.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "ACCOUNT-B-ID"},
    "Action": "sts:AssumeRole",
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }
}
```

Per ulteriori informazioni sui tipi di condizioni per MFA, consulta AWS chiavi di contesto della condizione globale, Operatori di condizione numerici e Operatore di condizione per verificare la presenza di chiavi di condizione .

Scegliendo tra GetSessionToken e AssumeRole

AWS STS fornisce due operazioni API che consentono agli utenti di trasmettere informazioni MFA: `GetSessionToken` e. `AssumeRole` L'operazione API che l'utente chiama per ottenere le credenziali di sicurezza temporanee dipende dallo scenario applicabile tra quelli descritti di seguito.

Usa **`GetSessionToken`** per gli scenari seguenti:

- Chiama le operazioni API che accedono alle risorse nello Account AWS stesso modo in cui l'utente IAM effettua la richiesta. Tieni presente che le credenziali temporanee di una `GetSessionToken` richiesta possono accedere alle operazioni IAM e AWS STS API solo se includi informazioni MFA nella richiesta di credenziali. Poiché le credenziali temporanee restituite da `GetSessionToken` includono le informazioni MFA, puoi verificare l'MFA nelle singole operazioni API effettuate tramite le credenziali.
- Accesso alle risorse protette con policy basate su risorse che includono una condizione MFA.

Lo scopo dell'operazione `GetSessionToken` è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare le operazioni di autenticazione.

Usa **`AssumeRole`** per gli scenari seguenti:

- Chiamata alle operazioni API che accedono alle risorse nello stesso Account AWS o in un account diverso. Le chiamate API possono includere qualsiasi IAM o API. AWS STS Per proteggere l'accesso, l'autenticazione MFA viene applicata quando l'utente assume il ruolo. Le credenziali temporanee restituite da `AssumeRole` non includono le informazioni MFA nel contesto, quindi non puoi verificare l'MFA nelle singole operazioni API. Per questo motivo, è necessario usare `GetSessionToken` per limitare l'accesso alle risorse protette da policy basate su risorse.

Le informazioni su come implementare questi scenari vengono fornite più avanti in questo documento.

Considerazioni importanti sull'accesso alle API protetto da MFA

È importante comprendere i seguenti aspetti della protezione MFA per le operazioni API:

- La protezione MFA è disponibile solo con le credenziali di sicurezza temporanee, che è possibile ottenere con `AssumeRole` o `GetSessionToken`.
- Non è possibile utilizzare l'accesso alle API protetto da MFA con credenziali. Utente root dell'account AWS

- Non è possibile usare l'accesso alle API protetto da MFA con chiavi di sicurezza U2F.
- Agli utenti federati non può essere assegnato un dispositivo MFA da utilizzare AWS con i servizi, quindi non possono AWS accedere alle risorse controllate dalla MFA. Vedi il punto successivo.
- Altre operazioni AWS STS API che restituiscono credenziali temporanee non supportano l'MFA. Per `AssumeRoleWithWebIdentity` e`AssumeRoleWithSAML`, l'utente è autenticato da un provider esterno e AWS non può determinare se tale provider abbia richiesto l'autenticazione MFA. Per `GetFederationToken`, l'autenticazione MFA non è necessariamente associata a un utente specifico.
- Analogamente, le credenziali a lungo termine (chiavi di accesso dell'utente IAM e chiavi di accesso dell'utente root) non possono essere usate con l'accesso alle API protetto da MFA perché non scadono.
- È possibile chiamare `AssumeRole` e `GetSessionToken` anche senza informazioni MFA. In tal caso, il chiamante riceve le credenziali di sicurezza temporanee, ma le informazioni di sessione per tali credenziali temporanee non indicano che l'utente ha eseguito l'autenticazione con MFA.
- Per stabilire la protezione MFA per le operazioni API, aggiungi condizioni MFA alle policy. Una policy deve includere la chiave di condizione `aws:MultiFactorAuthPresent` per implementare l'uso dell'MFA. Per la delega tra più account, la policy di attendibilità del ruolo deve includere la chiave di condizione.
- Quando consenti Account AWS a un altro utente di accedere alle risorse del tuo account, la sicurezza delle tue risorse dipende dalla configurazione dell'account fidato (l'altro account, non il tuo). Questo vale anche quando è richiesta la multi-factor authentication. Qualsiasi identità nell'account attendibile che dispone dell'autorizzazione per creare dispositivi MFA virtuali può creare un'attestazione MFA per soddisfare tale parte della policy di affidabilità del ruolo. Prima di consentire ai membri di un altro account di accedere alle tue AWS risorse che richiedono l'autenticazione a più fattori, devi assicurarti che il proprietario dell'account fidato segua le migliori pratiche di sicurezza. Ad esempio, l'account attendibile deve limitare l'accesso alle operazioni API sensibili, ad esempio le operazioni API di gestione dei dispositivi MFA, a identità attendibili specifiche.
- Se una policy include una condizione MFA, una richiesta viene negata se gli utenti non sono stati autenticati tramite MFA oppure se forniscono una password TOTP o un identificatore di dispositivo MFA non valido.

Scenario: Protezione MFA per la delega tra account

In questo scenario, desideri delegare l'accesso agli utenti IAM in un altro account, ma solo se gli utenti sono autenticati con un dispositivo MFA AWS . Per ulteriori informazioni sulla delega tra più account, consulta Termini e concetti dei ruoli.

Immagina di avere un account A (l'account che determina l'attendibilità, proprietario della risorsa a cui è necessario accedere), con l'utente IAM Anaya, che ha l'autorizzazione di amministratore. Anaya desidera concedere l'accesso all'utente Richard nell'account B (l'account attendibile), ma vuole assicurarsi che Richard sia autenticato con MFA prima di poter assumere il ruolo.

1. Nell'account di fiducia A, Anaya crea un ruolo IAM denominato `CrossAccountRole` e imposta come principale nella politica di fiducia del ruolo l'ID dell'account B. La politica di fiducia concede l'autorizzazione all'azione. AWS STS `AssumeRole` Anaya aggiunge inoltre una condizione MFA alla policy di trust, come nell'esempio seguente.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "ACCOUNT-B-ID"},
    "Action": "sts:AssumeRole",
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }
}
```

2. Anaya aggiunge una policy di autorizzazione al ruolo che specifica le attività consentite per il ruolo. La policy di autorizzazione per un ruolo con protezione MFA è uguale a qualsiasi altra policy di autorizzazione di un ruolo. L'esempio seguente mostra la policy aggiunta al ruolo da Anaya, che consente a un utente ipotetico di eseguire qualsiasi operazione Amazon DynamoDB sulla tabella `Books` nell'account A. Questa policy consente anche l'operazione `dynamodb:ListTables`, necessaria per eseguire operazioni nella console.

> **Note**
>
> La policy di autorizzazione non include una condizione MFA. È importante comprendere che l'autenticazione MFA viene usata solo per determinare se un utente può assumere tale ruolo. Una volta che l'utente ha assunto il ruolo, non vengono svolti ulteriori controlli MFA.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TableActions",
            "Effect": "Allow",
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:*:ACCOUNT-A-ID:table/Books"
        },
        {
            "Sid": "ListTables",
            "Effect": "Allow",
            "Action": "dynamodb:ListTables",
            "Resource": "*"
        }
    ]
}
```

3. Nell'account attendibile B, l'amministratore si assicura che l'utente IAM Richard sia configurato con un dispositivo AWS MFA e che conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.
4. Nell'account B, l'amministratore collega all'utente Richard (o un gruppo di cui è membro) la policy seguente, che gli permette di chiamare l'operazione `AssumeRole`. La risorsa è impostata sull'ARN del ruolo creato da Anaya nella fase 1. Osserva che questa policy non contiene una condizione MFA.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["sts:AssumeRole"],
    "Resource": ["arn:aws:iam::ACCOUNT-A-ID:role/CrossAccountRole"]
  }]
}
```

5. Nell'account B, Richard (o un'applicazione che Richard sta eseguendo) chiama `AssumeRole`. La chiamata dell'API include l'ARN del ruolo da assumere (`arn:aws:iam::ACCOUNT-A-ID:role/CrossAccountRole`), l'ID del dispositivo MFA e la password TOTP corrente che Richard ottiene dal suo dispositivo.

Quando Richard chiama`AssumeRole`, AWS determina se dispone di credenziali valide, incluso il requisito per l'MFA. In caso affermativo, Richard assume correttamente il ruolo e può eseguire qualsiasi operazione DynamoDB sulla tabella denominata `Books` nell'account A usando le credenziali temporanee del ruolo.

Per un esempio di programma che chiama `AssumeRole`, consulta AssumeRole Chiamate con autenticazione MFA.

Scenario: Protezione MFA per l'accesso alle operazioni API nell'account corrente

In questo scenario, dovresti assicurarti che un tuo utente Account AWS possa accedere alle operazioni API sensibili solo quando l'utente è autenticato utilizzando un dispositivo AWS MFA.

Immagina di avere un account A che contiene un gruppo di sviluppatori che devono usare le istanze EC2. In genere, gli sviluppatori possono usare le istanze, ma non hanno le autorizzazioni per le operazioni `ec2:StopInstances` e `ec2:TerminateInstances`. È opportuno limitare queste operazioni privilegiate "distruttive" a pochi utenti attendibili, quindi aggiungi la protezione MFA alla policy che permette queste operazioni Amazon EC2 sensibili.

In questo scenario, uno degli utenti attendibili è Sofía. L'utente Anaya è un amministratore nell'account A.

1. Anaya si assicura che Sofía sia configurata con un dispositivo AWS MFA e che Sofía conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.
2. Anaya crea un gruppo denominato `EC2-Admins` e aggiunge l'utente Sofía al gruppo.
3. Anaya collega la policy seguente al gruppo `EC2-Admins`. Questa policy concede agli utenti l'autorizzazione per chiamare le operazioni `StopInstances` e `TerminateInstances` di Amazon EC2 solo se l'utente ha eseguito l'autenticazione tramite MFA.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:StopInstances",
      "ec2:TerminateInstances"
    ],
```

```
    "Resource": ["*"],
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }]
}
```

4.
> **Note**
>
> Per rendere effettiva questa policy, gli utenti devono prima disconnettersi e quindi accedere nuovamente.

   Se l'utente Sofía deve arrestare o terminare un'istanza Amazon EC2, l'utente o un'applicazione da lei eseguita, chiama `GetSessionToken`. Questa operazione API passa l'ID o del dispositivo MFA e la password TOTP corrente che Sofía ottiene dal suo dispositivo.
5. L'utente Sofía (o un'applicazione che Sofía sta usando) usa le credenziali temporanee fornite da `GetSessionToken` per chiamare l'operazione `StopInstances` o `TerminateInstances` di Amazon EC2.

   Per un esempio di programma che chiama `GetSessionToken`, consulta [GetSessionToken Chiamate con autenticazione MFA] più avanti in questo documento.

Scenario: Protezione MFA per le risorse che hanno policy basate su risorse

In questo scenario, sei il proprietario di un bucket S3, una coda SQS o un argomento SNS. Vuoi assicurarti che tutti gli utenti Account AWS che accedono alla risorsa siano autenticati da un dispositivo MFA AWS .

Questo scenario illustra un modo per fornire la protezione MFA per più account senza richiedere agli utenti di assumere prima un ruolo. In tal caso, l'utente può accedere alla risorsa se vengono soddisfatte tre condizioni: l'utente deve essere autenticato mediante MFA, essere in grado di ottenere credenziali di sicurezza temporanee da `GetSessionToken` ed essere in un account ritenuto attendibile dalla policy della risorsa.

Immagina di avere l'account A e di creare un bucket S3. Desideri concedere l'accesso a questo bucket agli utenti che si trovano in diversi ambienti Account AWS, ma solo se tali utenti sono autenticati con MFA.

In questo scenario, l'utente Anaya è un amministratore nell'account A. L'utente Nikhil è un utente IAM nell'account C.

1. Nell'account A, Anaya crea un bucket denominato `Account-A-bucket`.
2. Anaya aggiunge la policy del bucket al bucket. La policy permette a qualsiasi utente in un account A, un account B o un account C di eseguire le operazioni `PutObject` e `DeleteObject` di Amazon S3 nel bucket. La policy include una condizione MFA.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"AWS": [
      "ACCOUNT-A-ID",
      "ACCOUNT-B-ID",
      "ACCOUNT-C-ID"
    ]},
    "Action": [
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": ["arn:aws:s3:::ACCOUNT-A-BUCKET-NAME/*"],
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }]
}
```

> **Note**
>
> Amazon S3 offre la funzionalità Cancellazione MFA solo per l'accesso all'account root. Puoi abilitare la funzionalità Cancellazione MFA di Amazon S3 quando imposti lo stato del controllo delle versioni del bucket. La funzionalità Cancellazione MFA di Amazon S3 non può essere applicata a un utente IAM e viene gestita indipendentemente dall'accesso alle API protetto da MFA. Un utente IAM con l'autorizzazione per eliminare un bucket non può eliminare un bucket quando la funzionalità Cancellazione MFA di Amazon S3 è abilitata. Per ulteriori informazioni sulla funzionalità Cancellazione MFA di Amazon S3, consulta Cancellazione MFA.

3. Nell'account C, un amministratore verifica che l'utente Nikhil sia configurato con un dispositivo MFA AWS e che conosca l'ID del dispositivo. ovvero il numero di serie se si tratta di un dispositivo MFA hardware o l'ARN del dispositivo se si tratta di un dispositivo MFA virtuale.

4. Nell'account C, Nikhil (o un'applicazione che lui sta eseguendo) chiama `GetSessionToken`. La chiamata include l'ID o l'ARN del dispositivo MFA e la password TOTP corrente che Nikhil ottiene dal suo dispositivo.
5. Nikhil (o un'applicazione che lui sta usando) usa le credenziali temporanee restituite da `GetSessionToken` per chiamare l'operazione `PutObject` di Amazon S3 per caricare un file in `Account-A-bucket`.

   Per un esempio di programma che chiama `GetSessionToken`, consulta GetSessionToken Chiamate con autenticazione MFA più avanti in questo documento.

   > **Note**
   >
   > Le credenziali temporanee che `AssumeRole` restituisce non funzionano in questo caso. Anche se l'utente è in grado di fornire informazioni MFA per assumere un ruolo, le credenziali temporanee restituite da `AssumeRole` non includono le informazioni MFA. Queste informazione sono necessarie per soddisfare la condizione MFA nella policy.

Codice di esempio: richiesta di credenziali con l'autenticazione a più fattori (MFA)

I seguenti esempi mostrano come chiamare le operazioni `GetSessionToken` e `AssumeRole` e passare i parametri di autenticazione MFA. Non è richiesta alcuna autorizzazione per chiamare `GetSessionToken`, ma è necessario disporre di una policy che permetta di chiamare `AssumeRole`. Le credenziali restituite vengono quindi utilizzate per elencare tutti i bucket S3 nell'account.

GetSessionToken Chiamate con autenticazione MFA

I seguenti esempi mostrano come chiamare `GetSessionToken` e passare le informazioni sull'autenticazione MFA. Le credenziali di sicurezza temporanee restituite dall'operazione `GetSessionToken` vengono quindi utilizzate per elencare tutti i bucket S3 nell'account.

La policy collegata all'utente che esegue questo codice (o a un gruppo in cui si trova utente) fornisce le autorizzazioni per le credenziali temporanee restituite. Per questo codice di esempio, la policy deve concedere all'utente l'autorizzazione per richiedere l'operazione `ListBuckets` di Amazon S3.

I seguenti esempi di codice mostrano come utilizzare`GetSessionToken`.

CLI

AWS CLI

Come ottenere un set di credenziali a breve termine per un'identità IAM

il comando `get-session-token` seguente recupera un set di credenziali a breve termine per l'identità IAM che esegue la chiamata. Le credenziali risultanti possono essere utilizzate per richieste in cui l'autenticazione a più fattori (MFA) è richiesta dalla policy. Le credenziali scadono 15 minuti dopo la loro generazione.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetSessionToken AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce un'**`Amazon.RuntimeAWSCredentials`**istanza contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere

credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzate i parametri - o - /. ProfileName AccessKey SecretKey

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 2: restituisce un'**Amazon.RuntimeAWSCredentials**istanza contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 3: restituisce un'**Amazon.RuntimeAWSCredentials**istanza contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. GetSessionTokenAWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un token di sessione passando un token MFA e utilizzalo per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
```

```
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta GetSessionToken AWSSDK for Python (Boto3) API Reference.

AssumeRole Chiamate con autenticazione MFA

I seguenti esempi mostrano come chiamare `AssumeRole` e passare le informazioni sull'autenticazione MFA. Le credenziali di sicurezza temporanee restituite da `AssumeRole` vengono quindi utilizzate per elencare tutti i bucket Amazon S3 nell'account.

Per ulteriori informazioni su questo scenario, consulta Scenario: Protezione MFA per la delega tra account.

I seguenti esempi di codice mostrano come utilizzare`AssumeRole`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;

namespace AssumeRoleExample
{
    class AssumeRole
    {
        /// <summary>
        /// This example shows how to use the AWS Security Token
        /// Service (AWS STS) to assume an IAM role.
        ///
        /// NOTE: It is important that the role that will be assumed has a
        /// trust relationship with the account that will assume the role.
        ///
        /// Before you run the example, you need to create the role you want to
        /// assume and have it trust the IAM account that will assume that role.
        ///
        /// See https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_create.html
        /// for help in working with roles.
        /// </summary>

        private static readonly RegionEndpoint REGION = RegionEndpoint.USWest2;

        static async Task Main()
        {
            // Create the SecurityToken client and then display the identity of
 the
            // default user.
            var roleArnToAssume = "arn:aws:iam::123456789012:role/
testAssumeRole";

            var client = new
 Amazon.SecurityToken.AmazonSecurityTokenServiceClient(REGION);

            // Get and display the information about the identity of the default
 user.
            var callerIdRequest = new GetCallerIdentityRequest();
            var caller = await client.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"Original Caller: {caller.Arn}");
```

```
            // Create the request to use with the AssumeRoleAsync call.
            var assumeRoleReq = new AssumeRoleRequest()
            {
                DurationSeconds = 1600,
                RoleSessionName = "Session1",
                RoleArn = roleArnToAssume
            };

            var assumeRoleRes = await client.AssumeRoleAsync(assumeRoleReq);

            // Now create a new client based on the credentials of the caller
 assuming the role.
            var client2 = new AmazonSecurityTokenServiceClient(credentials:
 assumeRoleRes.Credentials);

            // Get and display information about the caller that has assumed the
 defined role.
            var caller2 = await client2.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"AssumedRole Caller: {caller2.Arn}");
        }
    }
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
```

```
################################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

################################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
################################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary
 credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
```

```
  }

  while getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        ech o"Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AssumeRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::STS::assumeRole(const Aws::String &roleArn,
                             const Aws::String &roleSessionName,
                             const Aws::String &externalId,
                             Aws::Auth::AWSCredentials &credentials,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::STS::STSClient sts(clientConfig);
    Aws::STS::Model::AssumeRoleRequest sts_req;

    sts_req.SetRoleArn(roleArn);
    sts_req.SetRoleSessionName(roleSessionName);
    sts_req.SetExternalId(externalId);

    const Aws::STS::Model::AssumeRoleOutcome outcome = sts.AssumeRole(sts_req);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error assuming IAM role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Credentials successfully retrieved." << std::endl;
        const Aws::STS::Model::AssumeRoleResult result = outcome.GetResult();
        const Aws::STS::Model::Credentials &temp_credentials =
 result.GetCredentials();

        // Store temporary credentials in return argument.
        // Note: The credentials object returned by assumeRole differs
        // from the AWSCredentials object used in most situations.
        credentials.SetAWSAccessKeyId(temp_credentials.GetAccessKeyId());
        credentials.SetAWSSecretKey(temp_credentials.GetSecretAccessKey());
        credentials.SetSessionToken(temp_credentials.GetSessionToken());
    }
```

```
    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Come assumere un ruolo

Il comando `assume-role` seguente recupera un set di credenziali a breve termine per il ruolo IAM `s3-access-example`.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

Output:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-
access-example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8B
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che puoi utilizzare per l'autenticazione in AWS.

Per l'utilizzo della AWS CLI, è possibile impostare un profilo denominato associato a un ruolo. Quando utilizzi il profilo, la AWS CLI chiamerà assume-role e gestirà le credenziali per te. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM nella CLI nella AWS CLI User Guide AWS .

- Per i dettagli sull'API, consulta AWS CLI Command AssumeRoleReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sts.StsClient;
import software.amazon.awssdk.services.sts.model.AssumeRoleRequest;
import software.amazon.awssdk.services.sts.model.StsException;
import software.amazon.awssdk.services.sts.model.AssumeRoleResponse;
import software.amazon.awssdk.services.sts.model.Credentials;
import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;
import java.util.Locale;

/**
 * To make this code example work, create a Role that you want to assume.
 * Then define a Trust Relationship in the AWS Console. You can use this as an
 * example:
 *
 * {
 * "Version": "2012-10-17",
 * "Statement": [
 * {
 * "Effect": "Allow",
```

```
 * "Principal": {
 * "AWS": "<Specify the ARN of your IAM user you are using in this code
 * example>"
 * },
 * "Action": "sts:AssumeRole"
 * }
 * ]
 * }
 *
 * For more information, see "Editing the Trust Relationship for an Existing
 * Role" in the AWS Directory Service guide.
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AssumeRole {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleArn> <roleSessionName>\s

                Where:
                    roleArn - The Amazon Resource Name (ARN) of the role to
 assume (for example, rn:aws:iam::000008047983:role/s3role).\s
                    roleSessionName - An identifier for the assumed role session
 (for example, mysession).\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleArn = args[0];
        String roleSessionName = args[1];
        Region region = Region.US_EAST_1;
        StsClient stsClient = StsClient.builder()
                .region(region)
                .build();
```

```
        assumeGivenRole(stsClient, roleArn, roleSessionName);
        stsClient.close();
    }

    public static void assumeGivenRole(StsClient stsClient, String roleArn,
 String roleSessionName) {
        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();

            // Display the time when the temp creds expire.
            Instant exTime = myCreds.expiration();
            String tokenInfo = myCreds.sessionToken();

            // Convert the Instant to readable date.
            DateTimeFormatter formatter =
 DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
                    .withLocale(Locale.US)
                    .withZone(ZoneId.systemDefault());

            formatter.format(exTime);
            System.out.println("The token " + tokenInfo + "  expires on " +
 exTime);

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea il client.

```
import { STSClient } from "@aws-sdk/client-sts";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an AWS STS service client object.
export const client = new STSClient({ region: REGION });
```

Assumi il ruolo IAM.

```
import { AssumeRoleCommand } from "@aws-sdk/client-sts";

import { client } from "../libs/client.js";

export const main = async () => {
  try {
    // Returns a set of temporary security credentials that you can use to
    // access Amazon Web Services resources that you might not normally
    // have access to.
    const command = new AssumeRoleCommand({
      // The Amazon Resource Name (ARN) of the role to assume.
      RoleArn: "ROLE_ARN",
      // An identifier for the assumed role session.
      RoleSessionName: "session1",
      // The duration, in seconds, of the role session. The value specified
      // can range from 900 seconds (15 minutes) up to the maximum session
      // duration set for the role.
      DurationSeconds: 900,
    });
    const response = await client.send(command);
    console.log(response);
```

```
  } catch (err) {
    console.error(err);
  }
};
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
const AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

var roleToAssume = {
  RoleArn: "arn:aws:iam::123456789012:role/RoleName",
  RoleSessionName: "session1",
  DurationSeconds: 900,
};
var roleCreds;

// Create the STS service object
var sts = new AWS.STS({ apiVersion: "2011-06-15" });

//Assume Role
sts.assumeRole(roleToAssume, function (err, data) {
  if (err) console.log(err, err.stack);
  else {
    roleCreds = {
      accessKeyId: data.Credentials.AccessKeyId,
      secretAccessKey: data.Credentials.SecretAccessKey,
      sessionToken: data.Credentials.SessionToken,
    };
    stsGetCallerIdentity(roleCreds);
  }
});
```

```
//Get Arn of current identity
function stsGetCallerIdentity(creds) {
  var stsParams = { credentials: creds };
  // Create STS service object
  var sts = new AWS.STS(stsParams);

  sts.getCallerIdentity({}, function (err, data) {
    if (err) {
      console.log(err, err.stack);
    } else {
      console.log(data.Arn);
    }
  });
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla politica di accesso del ruolo assunto e dalla politica fornita (non è possibile utilizzare la politica fornita per concedere autorizzazioni superiori a quelle definite dalla politica di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella politica di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Assumi un ruolo IAM che richiede un token MFA e utilizza le credenziali temporanee per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_from_assumed_role_with_mfa(
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
```

```
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
MFA
                              device, this is an ARN.
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: "create-use-assume-role-scenario"
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
async fn assume_role(config: &SdkConfig, role_name: String, session_name:
 Option<String>) {
    let provider = aws_config::sts::AssumeRoleProvider::builder(role_name)
        .session_name(session_name.unwrap_or("rust_sdk_example_session".into()))
        .configure(config)
        .build()
        .await;

    let local_config = aws_config::from_env()
        .credentials_provider(provider)
        .load()
        .await;
    let client = Client::new(&local_config);
    let req = client.get_caller_identity();
    let resp = req.send().await;
    match resp {
        Ok(e) => {
            println!("UserID :               {}",
 e.user_id().unwrap_or_default());
            println!("Account:               {}",
 e.account().unwrap_or_default());
            println!("Arn    :               {}", e.arn().unwrap_or_default());
        }
        Err(e) => println!("{:?}", e),
    }
}
```

- Per i dettagli sulle API, consulta il riferimento AssumeRoleall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func assumeRole(role: IAMClientTypes.Role, sessionName: String)
                async throws -> STSClientTypes.Credentials {
    let input = AssumeRoleInput(
        roleArn: role.arn,
        roleSessionName: sessionName
    )
    do {
        let output = try await stsClient.assumeRole(input: input)

        guard let credentials = output.credentials else {
            throw ServiceHandlerError.authError
        }

        return credentials
    } catch {
        throw error
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Swift.

Trova credenziali inutilizzate AWS

Per aumentare la sicurezza del tuo account Account AWS, rimuovi le credenziali utente IAM (ovvero password e chiavi di accesso) che non sono necessarie. Ad esempio, quando gli utenti lasciano l'organizzazione o non hanno più bisogno di AWS accedervi, trovate le credenziali che stavano utilizzando e assicuratevi che non siano più operative. La soluzione ideale consiste nell'eliminare tutte le credenziali inutilizzate. Se l'utente dovesse averne bisogno in un secondo momento, potrai sempre ricrearle. Come minimo, dovresti modificare la password o disattivare le chiavi di accesso, per impedire l'accesso agli ex utenti.

Ovviamente, la definizione di inutilizzata è abbastanza ambigua, ma in genere si intende che una credenziale non è stata utilizzata per un determinato periodo di tempo.

Ricerca di password inutilizzate

Puoi utilizzarli AWS Management Console per visualizzare le informazioni sull'utilizzo delle password per i tuoi utenti. Se il numero di utenti è elevato, puoi utilizzare la console per scaricare un report delle credenziali, con informazioni sui tempi di utilizzo delle password della console. Puoi anche accedere alle informazioni dall'API AWS CLI o dall'API IAM.

Per individuare le password inutilizzate (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Console last sign-in (Ultimo accesso alla console) nella tabella degli utenti:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Select visible columns (Seleziona colonne visibili), seleziona Console last sign-in (Ultimo accesso alla console).
   c. Seleziona Confirm (Conferma) per tornare all'elenco degli utenti.
4. La colonna dell'ultimo accesso alla console mostra la data dell'ultimo accesso dell'utente AWS tramite la console. Mediante queste informazioni puoi trovare gli utenti le cui password non sono state utilizzate per un determinato periodo di tempo. La colonna mostra Never (Mai) per gli utenti che non hanno mai usato le password per accedere. None (Nessuna) indica gli utenti senza password. Le password non utilizzate di recente potrebbero essere candidate ideali per la rimozione.

> **Important**
>
> A causa di un problema di servizio, i dati sull'ultimo utilizzo della password non includono il periodo compreso fra le 22.50 (PDT) del 3 maggio 2018 e le 14.08 (PDT) del 23 maggio 2018. Ciò influisce sulle date dell'ultimo accesso mostrate nella console IAM e sulle date dell'ultimo utilizzo della password nel rapporto sulle credenziali IAM e restituite dall'GetUser operazione API. Se un utente ha effettuato l'accesso durante il periodo

interessato, l'ultima data di utilizzo di password visualizzata sarà quella relativa all'ultimo utilizzo prima del 3 maggio 2018. Per gli utenti che hanno effettuato l'accesso dopo le 14.08 PDT del 23 maggio 2018, la data indicata sarà accurata.
Se utilizzi le informazioni relative all'ultima password utilizzata per identificare le credenziali non utilizzate da eliminare, ad esempio per eliminare gli utenti che non hanno effettuato l'accesso AWS negli ultimi 90 giorni, ti consigliamo di modificare la finestra di valutazione per includere date successive al 23 maggio 2018. In alternativa, se gli utenti utilizzano le chiavi di accesso per accedere a AWS livello di codice, è possibile fare riferimento alle ultime informazioni utilizzate sulla chiave di accesso, poiché sono accurate per tutte le date.

Per trovare le password inutilizzate scaricando il report delle credenziali (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, selezionare Credential report (Rapporto credenziali).
3. Selezionare Download Report (Scarica report) per scaricare un file CSV denominato `status_reports_<date>T<time>.csv`. La quinta colonna contiene la colonna `password_last_used` con le date o uno dei seguenti messaggi:
   - N/D: utenti a cui non è stata assegnata una password.
   - no_information: gli utenti che non hanno utilizzato la propria password da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per trovare le password inutilizzate (AWS CLI)

Per individuare le password non utilizzate, seguire il comando seguente:

- `aws iam list-users` restituisce un elenco di utenti, ciascuno con un valore `PasswordLastUsed`. Se il valore è mancante, l'utente non ha una password oppure la password non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per trovare le password (AWS API) non utilizzate

Per individuare le password non utilizzate, richiamare la seguente operazione:

- `ListUsers` restituisce una raccolta di utenti, ciascuno delle quali ha un valore `<PasswordLastUsed>`. Se il valore è mancante, l'utente non ha una password oppure la password non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (20 ottobre 2014).

Per informazioni sui comandi per scaricare il report delle credenziali, consultare Recupero dei report delle credenziali (AWS CLI).

Ricerca di chiavi di accesso inutilizzate

Puoi utilizzare il AWS Management Console per visualizzare le informazioni sull'utilizzo delle chiavi di accesso per i tuoi utenti. Se il numero di utenti è elevato, puoi utilizzare la console per scaricare un report delle credenziali per sapere quando gli utenti hanno utilizzato le loro chiavi di accesso. Puoi anche accedere alle informazioni dall'API AWS CLI o dall'API IAM.

Per trovare le chiavi di accesso inutilizzate (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Se necessario, aggiungere la colonna Access key last used (Ultimo utilizzo chiave di accesso) nella tabella degli utenti:
   a. Sopra la tabella all'estrema destra, selezionare l'icona delle impostazioni ( ).
   b. In Select visible columns (Seleziona colonne visibili), seleziona Access key last used (Ultima chiave d'accesso utilizzata).
   c. Seleziona Confirm (Conferma) per tornare all'elenco degli utenti.
4. La colonna Ultima chiave di accesso utilizzata mostra il numero di giorni trascorsi dall'ultimo accesso dell'utente a livello di AWS programmazione. Mediante queste informazioni puoi trovare le chiavi di accesso che non sono state utilizzate per un determinato periodo di tempo. Per gli utenti senza chiavi di accesso nella colonna è riportato –. Le chiavi di accesso non utilizzate di recente potrebbero essere candidate ideali per la rimozione.

Per trovare le chiavi di accesso inutilizzate scaricando il report delle credenziali (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel riquadro di navigazione, selezionare Credential Report (Rapporto credenziali).
3. Selezionare Download Report (Scarica report) per scaricare un file CSV denominato `status_reports_<date>T<time>.csv`. Le colonne da 11 a 13 contengono la data di ultimo utilizzo, la regione e le informazioni di servizio per la chiave di accesso 1. Le colonne da 16 a 18 contengono le stesse informazioni per la chiave di accesso 2. Il valore è N/D se l'utente non dispone di una chiave di accesso o se non l'ha utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (22 aprile 2015).

Per trovare le chiavi di accesso inutilizzate (AWS CLI)

Per individuare le chiavi di accesso non utilizzate, eseguire i comandi seguenti:

- `aws iam list-access-keys` restituisce informazioni sulle chiavi di accesso di un utente, tra cui `AccessKeyID`.
- `aws iam get-access-key-last-used` prende un ID chiave di accesso e restituisce informazioni, tra cui `LastUsedDate`, `Region` in cui la chiave di accesso è stata utilizzata e il `ServiceName` dell'ultimo servizio richiesto. Se `LastUsedDate` è mancante, la chiave di accesso non è stata utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (22 aprile 2015).

Per trovare chiavi di accesso (AWS API) non utilizzate

Per individuare le chiavi di accesso non utilizzate, richiamare le seguenti operazioni:

- `ListAccessKeys` restituisce un elenco di `AccessKeyID` valori per le chiavi di accesso associati all'utente specificato.
- `GetAccessKeyLastUsed` prende un ID chiave di accesso e restituisce una raccolta di valori. Sono incluse `LastUsedDate`, `Region` in cui la chiave di accesso è stato utilizzato e `ServiceName` dell'ultimo servizio richiesto. Se il valore è mancante, l'utente non dispone di una chiave di accesso oppure non l'ha utilizzata da quando IAM ha iniziato a monitorarne l'utilizzo (22 aprile 2015).

Per informazioni sui comandi per scaricare il report delle credenziali, consultare Recupero dei report delle credenziali (AWS CLI).

Recupero dei report delle credenziali per l' Account AWS

Puoi generare e scaricare un report delle credenziali che elenca tutti gli utenti presenti nel tuo account e lo stato delle loro diverse credenziali, tra cui le password, le chiavi di accesso e i dispositivi MFA. Puoi ottenere un rapporto sulle credenziali dagli AWS Management ConsoleAWS SDK e dagli strumenti a riga di comando o dall'API IAM.

Puoi utilizzare i report delle credenziali nei tuoi controlli e strategie di conformità. È possibile utilizzare il report per controllare gli effetti dei requisiti del ciclo di vita delle credenziali, come ad esempio la password e gli aggiornamenti della chiave di accesso. Puoi fornire il report a un revisore esterno o concedere le autorizzazioni a un'entità di controllo in modo che possa scaricare il report direttamente.

Puoi generare un report delle credenziali con una frequenza di una volta ogni quattro ore. Quando richiedi un report, IAM verifica innanzitutto se Account AWS è stato generato un report per il nelle ultime quattro ore. In questo caso, viene scaricato il report più recente. Se il report più recente per l'account è più vecchio di quattro ore oppure se non ci sono report precedenti per l'account, IAM genera e scarica un nuovo report.

Argomenti

Autorizzazioni richieste

Per creare e scaricare i report sono necessarie le seguenti autorizzazioni:

- Per creare un report delle credenziali: `iam:GenerateCredentialReport`
- Per scaricare il report: `iam:GetCredentialReport`

Comprendere il formato del report

I file dei report delle credenziali sono formattati con una virgola come separatore (CSV). Puoi aprire i file CSV con i comuni software per fogli di calcolo per eseguire l'analisi, oppure è possibile creare un'applicazione che riceva il file CSV in modo programmatico ed esegua un'analisi personalizzata.

Il file CSV contiene le seguenti colonne:

Utente

Il nome semplice dell'utente.

arn

L'Amazon Resource Name (ARN) dell'utente. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM.

user_creation_time

La data e l'ora in cui l'utente è stato creato, nel formato data/ora ISO 8601.

password_enabled

Quando l'utente ha una password, questo valore è `TRUE`. Altrimenti è`FALSE`. Il valore per Utente root dell'account AWS è always`not_supported`.

password_last_used

La data e l'ora in cui la password dell'utente Utente root dell'account AWS o dell'utente è stata utilizzata l'ultima volta per accedere a un AWS sito Web, nel formato data-ora ISO 8601. AWS i siti Web che registrano l'ora dell'ultimo accesso di un utente sono i AWS Management Console, i Forum di AWS discussione e il AWS Marketplace. Quando una password è utilizzata più volte in un intervallo di 5 minuti, viene registrato in questo campo solo il primo utilizzo.

- Il valore in questo campo è `no_information` in questi casi:
  - La password dell'utente non è mai stata utilizzata.
  - Non sono previsti dati di accesso associati alla password, come ad esempio quando la password dell'utente non è stata utilizzata dopo che IAM ha iniziato a monitorare queste informazioni il 20 ottobre 2014.
- Il valore di questo campo è `N/A` (non applicabile) quando l'utente non ha una password.

> **Important**
>
> A causa di un problema di servizio, i dati sull'ultimo utilizzo della password non includono il periodo compreso fra le 22.50 (PDT) del 3 maggio 2018 e le 14.08 (PDT) del 23 maggio 2018. Ciò influisce sulle date dell'ultimo accesso mostrate nella console IAM e sulle date dell'ultimo utilizzo della password nel rapporto sulle credenziali IAM e restituite dall'GetUser operazione API. Se un utente ha effettuato l'accesso durante il periodo interessato, l'ultima

data di utilizzo di password visualizzata sarà quella relativa all'ultimo utilizzo prima del 3 maggio 2018. Per gli utenti che hanno effettuato l'accesso dopo le 14.08 PDT del 23 maggio 2018, la data indicata sarà accurata.
Se utilizzi le informazioni relative all'ultima password utilizzata per identificare le credenziali non utilizzate da eliminare, ad esempio per eliminare gli utenti che non hanno effettuato l'accesso AWS negli ultimi 90 giorni, ti consigliamo di modificare la finestra di valutazione per includere date successive al 23 maggio 2018. In alternativa, se gli utenti utilizzano le chiavi di accesso per accedere a AWS livello di codice, è possibile fare riferimento alle ultime informazioni utilizzate sulla chiave di accesso, poiché sono accurate per tutte le date.

password_last_changed

La data e ora in cui la password dell'utente è stata impostata per l'ultima volta nel formato data/ora ISO 8601. Se l'utente non ha una password, il valore di questo campo è `N/A` (non applicabile). Il valore per Account AWS (root) è sempre. `not_supported`

password_next_rotation

Quando l'account ha una policy della password che richiede la rotazione della password, questo campo contiene la data e l'ora, nel formato data/ora ISO 8601, quando all'utente è richiesto di impostare una nuova password. Il valore per Account AWS (root) è sempre`not_supported`.

mfa_active

Quando un dispositivo Multi-Factor Authentication (MFA) è stato abilitato per l'utente, il valore è `TRUE`. In caso contrario è `FALSE`.

access_key_1_active

Quando l'utente ha una chiave di accesso e lo stato della chiave di accesso è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`.

access_key_1_last_rotated

La data e l'ora nel formato data/ora ISO 8601, della creazione della chiave di accesso dell'utente o dell'ultima modifica. Se l'utente non ha una chiave di accesso attiva, il valore in questo campo è `N/A` (non applicabile).

access_key_1_last_used_date

La data e l'ora, nel formato data/ora ISO 8601, in cui la chiave di accesso dell'utente è stata recentemente utilizzata per firmare una richiesta API AWS . Quando una chiave di accesso è

utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso non è stata utilizzata dopo che IAM ha iniziato a monitorarne le informazioni (22 aprile 2015).

access_key_1_last_used_region

La regione AWS in cui la chiave di accesso è stata utilizzata più recentemente. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).
- L'ultimo servizio utilizzato non è specifico della regione, come ad esempio Amazon S3.

access_key_1_last_used_service

Il AWS servizio a cui è stato effettuato l'ultimo accesso con la chiave di accesso. Il valore in questo campo utilizza lo spazio dei nomi del servizio, ad esempio `s3` per Amazon S3 e `ec2` per Amazon EC2. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso.
- La chiave di accesso non è mai stata utilizzata.
- La chiave di accesso è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

access_key_2_active

Quando l'utente ha una chiave di accesso secondaria e lo stato della chiave di accesso secondaria è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`.

> **Note**
>
> Gli utenti possono avere fino a due chiavi di accesso: ciò semplifica la rotazione, consentendo di aggiornare prima la chiave e successivamente di eliminare la chiave precedente. Per ulteriori informazioni sull'aggiornamento delle chiavi di accesso, consulta la pagina Aggiornamento delle chiavi di accesso.

access_key_2_last_rotated

La data e l'ora, espresse in formato data/ora ISO 8601, di creazione o ultima modifica della seconda chiave di accesso dell'utente. Se l'utente non ha una chiave di accesso secondaria attiva, il valore in questo campo è `N/A` (non applicabile).

access_key_2_last_used_date

La data e l'ora, in formato data-ora ISO 8601, in cui la seconda chiave di accesso dell'utente è stata utilizzata l'ultima volta per firmare una AWS richiesta API. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo.

Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

access_key_2_last_used_region

La regione AWS in cui la chiave di accesso secondaria è stata utilizzata più recentemente. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

- L'ultimo servizio utilizzato non è specifico della regione, come ad esempio Amazon S3.

access_key_2_last_used_service

Il AWS servizio a cui è stato effettuato l'ultimo accesso con la seconda chiave di accesso dell'utente. Il valore in questo campo utilizza lo spazio dei nomi del servizio, ad esempio `s3` per Amazon S3 e `ec2` per Amazon EC2. Quando una chiave di accesso è utilizzata più volte in un intervallo di 15 minuti, viene registrato in questo campo solo il primo utilizzo. Il valore in questo campo è `N/A` (non applicabile) in questi casi:

- L'utente non ha una chiave di accesso secondaria.
- La chiave di accesso secondaria dell'utente non è mai stata utilizzata.
- La chiave di accesso secondaria dell'utente è stata utilizzata l'ultima volta prima che IAM iniziasse a monitorarne le informazioni (22 aprile 2015).

cert_1_active

Quando l'utente ha un certificato di firma X.509 e lo stato del certificato è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`.

cert_1_last_rotated

La data e l'ora nel formato data/ora ISO 8601 della creazione del certificato di firma dell'utente o dell'ultima modifica. Se l'utente non ha un certificato di firma attivo, il valore in questo campo è `N/A` (non applicabile).

cert_2_active

Quando l'utente ha un certificato di firma X.509 secondario e lo stato del certificato è `Active`, questo valore è `TRUE`. In caso contrario è `FALSE`.

> **Note**
>
> Gli utenti possono avere fino a due certificati di firma X.509, per rendere più semplice la rotazione del certificato.

cert_2_last_rotated

La data e l'ora nel formato data/ora ISO 8601 della creazione del certificato di firma secondario dell'utente o dell'ultima modifica. Se l'utente non ha un certificato di firma secondario attivo, il valore in questo campo è `N/A` (non applicabile).

Recupero dei report delle credenziali (Console)

Puoi utilizzare il AWS Management Console per scaricare un rapporto sulle credenziali come file con valori separati da virgole (CSV).

Per scaricare un report delle credenziali (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, selezionare Credential report (Rapporto credenziali).
3. Scegliere Download Report (Scarica report).

Recupero dei report delle credenziali (AWS CLI)

Per scaricare un report sulle credenziali (AWS CLI)

1. Genera un rapporto sulle credenziali. AWS memorizza un singolo report. Se esiste già un report, la generazione di un report sulle credenziali sovrascrive il report precedente. `aws iam generate-credential-report`
2. Visualizza l'ultimo report generato: `aws iam get-credential-report`

Ottenere report sulle credenziali (AWS API)

Per scaricare un rapporto sulle credenziali (API)AWS

1. Genera un rapporto sulle credenziali. AWS memorizza un singolo report. Se esiste già un report, la generazione di un report sulle credenziali sovrascrive il report precedente. `GenerateCredentialReport`
2. Visualizza l'ultimo report generato: `GetCredentialReport`

Usare IAM con CodeCommit: credenziali Git, chiavi SSH e AWS chiavi di accesso

CodeCommit è un servizio di controllo delle versioni gestito che ospita repository Git privati nel AWS cloud. Per utilizzarlo CodeCommit, configuri il tuo client Git per comunicare con i CodeCommit

repository. Come parte di questa configurazione, fornisci credenziali IAM che CodeCommit possono essere utilizzate per autenticarti. IAM supporta tre tipi CodeCommit di credenziali:

- Credenziali Git, una coppia di nome utente e password generata da IAM che puoi usare per comunicare con i CodeCommit repository tramite HTTPS.
- Chiavi SSH, una coppia di chiavi pubblica-privata generata localmente che puoi associare al tuo utente IAM per comunicare con i CodeCommit repository tramite SSH.
- AWS chiavi di accesso, che puoi utilizzare con l'helper per le credenziali incluso nella confezione per comunicare con i AWS CLI repository tramite HTTPS. CodeCommit

> **Note**
>
> Non è possibile utilizzare le chiavi SSH o le credenziali Git per accedere ai repository in un altro account AWS . Per informazioni su come configurare l'accesso ai CodeCommit repository per utenti e gruppi IAM in un altro archivio Account AWS, consulta Configurare l'accesso tra account a un AWS CodeCommit repository utilizzando i ruoli nella Guida per l'utente.AWS CodeCommit

Per ulteriori informazioni su ciascuna opzione, consultare le sezioni seguenti.

Usa credenziali Git e HTTPS con CodeCommit (consigliato)

Con le credenziali Git, generi una coppia statica di nome utente e password per il tuo utente IAM e utilizzi tali credenziali per le connessioni HTTPS. Puoi utilizzare queste credenziali anche con qualsiasi strumento di terza parte o ambiente di sviluppo integrato (IDE) che supporta le credenziali Git statiche.

Poiché queste credenziali sono universali per tutti i sistemi operativi supportati e compatibili con la maggior parte dei sistemi di gestione delle credenziali, ambienti di sviluppo e altri strumenti di sviluppo software, questo è il metodo consigliato. Puoi reimpostare la password per le credenziali Git in qualsiasi momento. Puoi anche rendere le credenziali inattive o eliminarle se non ne hai più bisogno.

> **Note**
>
> Non è possibile selezionare il nome utente e la password per le credenziali Git. IAM genera queste credenziali per aiutarti a garantire che soddisfino gli standard di sicurezza AWS

e proteggano gli archivi in. CodeCommit Puoi scaricare le credenziali una sola volta, nel momento in cui vengono generate. Assicurati di salvare le credenziali in una posizione sicura. Se necessario, puoi reimpostare la password in qualsiasi momento, ma questa operazione invalida le connessioni configurate con la password precedente. Devi riconfigurare le connessioni in modo che utilizzino la nuova password per poterti connettere nuovamente.

Per ulteriori informazioni, consultare i seguenti argomenti:

- Per creare un utente IAM, consulta Creare un utente IAM nel tuo Account AWS.
- Per generare e utilizzare credenziali Git con CodeCommit, consulta Per gli utenti HTTPS che utilizzano le credenziali Git nella Guida per l'AWS CodeCommit utente.

> **Note**
>
> La modifica del nome di un utente IAM dopo la generazione delle credenziali Git non comporta la modifica del nome utente delle credenziali. Il nome utente e la password rimangono invariati e validi.

Per aggiornare le credenziali specifiche del servizio

1. Creare un secondo set di credenziali specifico del servizio in aggiunta al set attualmente in uso.
2. Aggiornare tutte le applicazioni in modo da utilizzare il nuovo set di credenziali e confermare che le applicazioni funzionino.
3. Cambiare lo stato delle credenziali originali in "Inactive" (Non attivo).
4. Verificare che tutte le applicazioni funzionino ancora.
5. Eliminare le credenziali specifiche del servizio non attive.

Usa chiavi SSH e SSH con CodeCommit

Con le connessioni SSH, crei file di chiave pubblici e privati sulla tua macchina locale che Git e Git CodeCommit utilizzano per l'autenticazione SSH. La chiave pubblica va associata all'utente IAM e la chiave privata va archiviata nel computer locale. Per ulteriori informazioni, consultare i seguenti argomenti:

- Per creare un utente IAM, consulta Creare un utente IAM nel tuo Account AWS.
- Per creare una chiave pubblica SSH e associarla a un utente IAM consulta Per le connessioni SSH su Linux, macOS o Unix oppure Per le connessioni SSH su Windows nella Guida per l'utente di AWS CodeCommit .

> **Note**
>
> La chiave pubblica deve essere codificata in formato ssh-rsa o formato PEM. La lunghezza in bit minima della chiave pubblica è di 2.048 bit e la lunghezza massima è di 16.384 bit. Questo valore è separato dalla dimensione del file da caricare. Ad esempio, è possibile generare una chiave a 2.048 bit e il file PEM risultante è lungo 1.679 byte. Se fornisci la chiave pubblica in un altro formato o dimensione, verrà visualizzato un messaggio di errore indicante che il formato non è valido.

Usa HTTPS con l'helper per le AWS CLI credenziali e CodeCommit

In alternativa alle connessioni HTTPS con credenziali Git, puoi consentire a Git di utilizzare una versione con firma crittografica delle tue credenziali utente IAM o del ruolo dell'istanza Amazon EC2 ogni volta che Git deve AWS autenticarsi per interagire con i repository. CodeCommit Questo è l'unico metodo di connessione per i CodeCommit repository che non richiede un utente IAM. Inoltre, questo è il solo metodo che funziona con l'accesso federato e le credenziali temporanee. Per ulteriori informazioni, consultare i seguenti argomenti:

- Per ulteriori informazioni sull'accesso federato, consultare Provider di identità e federazione e Fornire l'accesso a utenti autenticati esternamente (federazione delle identità).
- Per ulteriori informazioni sulle credenziali temporanee, consulta Credenziali di sicurezza temporanee in IAM e Accesso temporaneo ai CodeCommit repository.

L'helper per AWS CLI le credenziali non è compatibile con altri sistemi di supporto per le credenziali, come Keychain Access o Windows Credential Management. Quando configuri le connessioni HTTPS con l'assistente credenziali, devi tenere presenti ulteriori considerazioni. Per ulteriori informazioni, consulta Per le connessioni HTTPS su Linux, macOS o Unix con il AWS CLI Credential Helper o Connessioni HTTPS su Windows con il Credential Helper nella Guida per l' AWS CLI utente.AWS CodeCommit

Utilizzo di IAM con Amazon Keyspaces (per Apache Cassandra)

Amazon Keyspaces (per Apache Cassandra) è un servizio di database gestito, scalabile, ad alta disponibilità e compatibile con Apache Cassandra. Puoi accedere ad Amazon Keyspaces tramite o AWS Management Console programmaticamente. Per accedere ad Amazon Keyspaces a livello di programmazione con credenziali specifiche del servizio, puoi utilizzare `cqlsh` o i driver Cassandra open source. Le credenziali specifiche del servizio includono un nome utente e una password come quelli che Cassandra utilizza per l'autenticazione e la gestione degli accessi. Puoi avere un massimo di due set di credenziali specifiche del servizio per ogni servizio supportato per utente.

Per accedere ad Amazon Keyspaces in modo programmatico con le chiavi di AWS accesso, puoi utilizzare l' AWS SDK, il AWS Command Line Interface (AWS CLI) o i driver Cassandra open source con il plug-in SigV4. Per ulteriori informazioni, consulta la sezione Connessione a livello di programmazione ad Amazon Keyspaces nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

> **Note**
>
> Se prevedi di interagire con Amazon Keyspaces solo tramite la console, non è necessario generare credenziali specifiche del servizio. Per ulteriori informazioni, consulta la sezione Accesso ad Amazon Keyspaces (per Apache Cassandra) nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Per ulteriori informazioni sulle autorizzazioni richieste per accedere ad Amazon Keyspaces, consulta Esempi di policy basate su identità per Amazon Keyspaces (per Apache Cassandra) nella Guida per gli sviluppatori di Amazon Keyspaces (per Apache Cassandra).

Generazione delle credenziali Amazon Keyspaces (console)

Puoi utilizzare AWS Management Console per generare le credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Come generare credenziali specifiche del servizio Amazon Keyspaces (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Users (Utenti) quindi selezionare il nome dell'utente che richiede le credenziali.

3. Nella scheda Credenziali di sicurezza in Credenziali per Amazon Keyspaces (per Apache Cassandra), scegli Genera credenziali.
4. Le credenziali specifiche del servizio sono ora disponibili. Questa è l'unica volta in cui è possibile visualizzare o scaricare la password. Non puoi recuperarla successivamente. Tuttavia, è possibile reimpostare la password in qualsiasi momento. Salva l'utente e la password in una posizione sicura, perché ne avrai bisogno in un secondo momento.

Generazione delle credenziali di Amazon Keyspaces (AWS CLI)

Puoi utilizzare AWS CLI per generare le credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Come generare credenziali specifiche del servizio Amazon Keyspaces (AWS CLI)

- Utilizza il seguente comando:
  - era iam create-service-specific-credential

Generazione di credenziali Amazon Keyspaces (API)AWS

Puoi utilizzare l' AWS API per generare credenziali Amazon Keyspaces (per Apache Cassandra) per i tuoi utenti IAM.

Per generare credenziali specifiche del servizio Amazon Keyspaces (API)AWS

- Completare la seguente operazione:
  - CreateServiceSpecificCredential

Gestione dei certificati server in IAM

Per abilitare le connessioni HTTPS al tuo sito Web o alla tua applicazione in AWS, devi disporre di un certificato server SSL/TLS. Per i certificati in una regione supportata da AWS Certificate Manager (ACM), consigliamo di utilizzare ACM per effettuare il provisioning, la gestione e la distribuzione dei certificati server. Nelle regioni non supportate, è necessario utilizzare IAM come gestore di certificati. Per informazioni sulle regioni supportate da ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

ACM è lo strumento preferito per il provisioning, la gestione e la distribuzione dei certificati del server. Con ACM puoi richiedere un certificato o distribuire un certificato ACM o esterno esistente alle risorse. AWS I certificati forniti da ACM sono gratuiti e vengono automaticamente rinnovati. In una regione supportata è possibile utilizzare ACM per gestire i certificati server dalla console o a livello di programmazione. Per ulteriori informazioni sull'utilizzo di ACM, consulta la Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni su come richiedere un certificato ACM, consulta Richiesta di un certificato pubblico o Richiesta di un certificato privato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager .

Utilizza IAM come gestore di certificati solo quando è necessario il supporto alle connessioni HTTPS in una regione che non è supportata da ACM. IAM crittografa in modo sicuro le chiavi private e archivia la versione crittografata nella memoria dei certificati SSL di IAM. IAM supporta la distribuzione di certificati server in tutte le regioni, ma è necessario ottenere il certificato da un provider esterno per utilizzarlo con. AWS Non è possibile caricare un certificato ACM in IAM. Inoltre, non è possibile gestire i certificati dalla console IAM.

Per ulteriori informazioni sul caricamento di certificati di terze parti in IAM, consulta i seguenti argomenti.

Indice

Caricamento di un certificato server (API)AWS

Per caricare un certificato del server in IAM, è necessario fornire il certificato e la chiave privata corrispondente. Quando il certificato non è autofirmato, è necessario fornire anche una catena di certificati. (La catena di certificati non necessaria se si carica un certificato autofirmato). Prima di caricare un certificato, assicurarsi di disporre di tutti questi elementi e di soddisfare i seguenti criteri:

- Il certificato deve essere valido al momento del caricamento. Non è possibile caricare un certificato prima dell'inizio del periodo di validità `NotBefore` o dopo la data di scadenza (la data `NotAfter` del certificato).
- La chiave di accesso non deve essere crittografata. Non è possibile caricare una chiave di accesso privata protetta da password o da passphrase. Per informazioni sulla decodifica di una chiave privata crittografata, consultare Risoluzione dei problemi.
- Il certificato, la chiave privata e la catena di certificati devono tutti essere codificati con PEM. Per informazioni sulla conversione di tali elementi in formato PEM, consultare Risoluzione dei problemi.

Per utilizzare l'API IAM per caricare un certificato, invia una UploadServerCertificaterichiesta. L'esempio seguente mostra come eseguire questa operazione con l'AWS Command Line Interface (AWS CLI). L'esempio presuppone quanto segue:

- Il certificato con codifica PEM è archiviato in un file denominato `Certificate.pem`.
- La catena di certificati con codifica PEM è archiviata in un file denominato `CertificateChain.pem`.
- La chiave privata non crittografata con codifica PEM è archiviata in un file denominato `PrivateKey.pem`.
- (Facoltativo) Desideri applicare un tag al certificato del server con una coppia chiave-valore. Ad esempio, è possibile aggiungere la chiave tag `Department` e il valore tag `Engineering` per facilitare l'identificazione e l'organizzazione dei certificati.

Per utilizzare il seguente comando esemplificativo, sostituisci questi nomi di file con il tuo. Sostituiscilo *`ExampleCertificate`*con un nome per il certificato caricato. Se desideri etichettare il certificato, sostituisci la coppia chiave-valore *`ExampleKey`*e *`ExampleValue`*tag con i tuoi valori. Digitare il comando su una linea continua. L'esempio seguente include interruzioni di linea e spazi aggiuntivi per agevolare la lettura.

```
aws iam upload-server-certificate --server-certificate-name ExampleCertificate
                                  --certificate-body file://Certificate.pem
                                  --certificate-chain file://CertificateChain.pem
                                  --private-key file://PrivateKey.pem
                                  --tags '{"Key": "ExampleKey", "Value":
 "ExampleValue"}'
```

Quando il comando precedente viene completato, restituisce i metadati relativi al certificato caricati, tra cui il relativo Amazon Resource Name (ARN), il nome descrittivo, l'identificatore (ID), la data di scadenza, i tag e molte altre informazioni.

> **Note**
>
> Se stai caricando un certificato server da utilizzare con Amazon CloudFront, devi specificare un percorso utilizzando l'`--path`opzione. Il percorso deve iniziare con `/cloudfront` e devono includere una barra finale (ad esempio, `/cloudfront/test/`).

Per utilizzare l'opzione AWS Tools for Windows PowerShell per caricare un certificato, usa ServerCertificatepublish-IAM.

Recupero di un certificato server (API)AWS

Per utilizzare l'API IAM per recuperare un certificato, invia una richiesta. GetServerCertificate L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI. Sostituiscilo *`ExampleCertificate`*con il nome del certificato da recuperare.

```
aws iam get-server-certificate --server-certificate-name ExampleCertificate
```

Quando il comando precedente viene completato, restituisce il certificato, la catena di certificati (se è stata caricata) e i metadati sul certificato.

> **Note**
>
> Non è possibile scaricare o recuperare una chiave privata da IAM dopo averla caricata.

Per utilizzare il AWS Tools for Windows PowerShell per recuperare un certificato, usa Get-IAM. ServerCertificate

Elenco dei certificati del server (API)AWS

Per utilizzare l'API IAM per elencare i certificati del server caricati, invia una ListServerCertificatesrichiesta. L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

```
aws iam list-server-certificates
```

Quando il comando precedente ha esito positivo, restituisce un elenco che contiene metadati relativi a ciascun certificato.

Per utilizzare per AWS Tools for Windows PowerShell elencare i certificati del server caricati, usa Get-IAM ServerCertificates.

Applicazione e rimozione di tag dei certificati server (API AWS)

Puoi allegare tag alle risorse IAM per organizzare e controllare l'accesso ad esse. Per utilizzare l'API IAM per etichettare un certificato server esistente, invia una TagServerCertificaterichiesta. L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

```
aws iam tag-server-certificate --server-certificate-name ExampleCertificate
                               --tags '{"Key": "ExampleKey", "Value":
 "ExampleValue"}'
```

Quando il comando precedente viene completato in modo corretto, non viene restituito alcun output.

Per utilizzare l'API IAM per rimuovere i tag da un certificato server, invia una UntagServerCertificaterichiesta. L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

```
aws iam untag-server-certificate --server-certificate-name ExampleCertificate
                               --tag-keys ExampleKeyName
```

Quando il comando precedente viene completato in modo corretto, non viene restituito alcun output.

Rinominare un certificato server o aggiornarne il percorso (API)AWS

Per utilizzare l'API IAM per rinominare un certificato del server o aggiornarne il percorso, invia una UpdateServerCertificaterichiesta. L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

Per utilizzare il seguente comando di esempio, sostituire i nomi dei certificati precedenti e nuovi e il percorso del certificato e digitare il comando su una riga continua. L'esempio seguente include interruzioni di linea e spazi aggiuntivi per agevolare la lettura.

```
aws iam update-server-certificate --server-certificate-name ExampleCertificate
```

```
                                    --new-server-certificate-name CloudFrontCertificate
                                    --new-path /cloudfront/
```

Quando il comando precedente ha esito positivo, non viene restituito alcun output.

Per utilizzare il AWS Tools for Windows PowerShell per rinominare un certificato del server o aggiornarne il percorso, usa update-IAM. ServerCertificate

Eliminazione di un certificato del server (API)AWS

Per utilizzare l'API IAM per eliminare un certificato del server, invia una DeleteServerCertificaterichiesta. L'esempio seguente mostra come eseguire questa operazione con l' AWS CLI.

Per utilizzare il seguente comando di esempio, sostituiscilo *ExampleCertificate*con il nome del certificato da eliminare.

```
aws iam delete-server-certificate --server-certificate-name ExampleCertificate
```

Quando il comando precedente ha esito positivo, non viene restituito alcun output.

Per utilizzare il AWS Tools for Windows PowerShell per eliminare un certificato del server, usa Remove-IAM ServerCertificate.

Risoluzione dei problemi

Prima di poter caricare un certificato in IAM, è necessario assicurarsi che il certificato, la chiave privata e la catena di certificati dispongano tutti della codifica PEM. È inoltre necessario assicurarsi che la chiave privata non sia crittografata. Fare riferimento agli esempi riportati di seguito.

Example Esempio di certificato con codifica PEM

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Example Esempio di chiave privata con codifica PEM, non crittografata

```
-----BEGIN RSA PRIVATE KEY-----
Base64-encoded private key
-----END RSA PRIVATE KEY-----
```

Example Esempio di catena di certificati con codifica PEM

Una catena di certificati contiene uno o più certificati. Puoi utilizzare un editor di testo, il comando di copia in Windows, oppure il comando Linux cat per concatenare i tuoi file del certificato in una catena. Quando includi più certificati, ogni certificato deve certificare il certificato precedente. Puoi farlo concatenando i certificati, incluso il certificato CA radice per ultimo.

L'esempio seguente contiene tre certificati, ma la catena di certificati può contenerne un numero maggiore o minore di certificati.

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Se questi elementi non sono nel formato corretto per il caricamento in IAM, puoi utilizzare OpenSSL per convertirli nel formato corretto.

Per convertire un certificato o una catena di certificati da DER a PEM

Utilizzare il comando OpenSSL x509, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *Certificate.der* con il nome del file che contiene il certificato con codifica DER. Sostituire *Certificate.pem* con il nome preferito del file di output per contenere il certificato con codifica PEM.

```
openssl x509 -inform DER -in Certificate.der -outform PEM -out Certificate.pem
```

Per convertire una chiave privata da DER a PEM

Utilizzare il comando OpenSSL rsa, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *PrivateKey.der* con il nome del file che contiene la chiave privata con codifica DER. Sostituire *PrivateKey.pem* con il nome preferito del file di output per contenere la chiave privata con codifica PEM.

```
openssl rsa -inform DER -in PrivateKey.der -outform PEM -out PrivateKey.pem
```

Per decrittografare una chiave privata crittografata (rimuovere la password o la passphrase)

Utilizzare il comando OpenSSL rsa, come nell'esempio seguente. Per utilizzare il seguente comando di esempio, sostituire *EncryptedPrivateKey.pem* con il nome del file che contiene la chiave privata crittografata. Sostituire *PrivateKey.pem* con il nome preferito del file di output per contenere la chiave privata con codifica PEM non crittografata.

```
openssl rsa -in EncryptedPrivateKey.pem -out PrivateKey.pem
```

Per convertire un bundle di certificati da PKCS # 12 (PFX) a PEM

Utilizzare il comando OpenSSL pkcs12, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *CertificateBundle.p12* con il nome del file che contiene il bundle di certificati con codifica PKCS#12. Sostituire *CertificateBundle.pem* con il nome preferito del file di output per contenere il bundle di certificati con codifica PEM.

```
openssl pkcs12 -in CertificateBundle.p12 -out CertificateBundle.pem -nodes
```

Per convertire un bundle di certificati da PKCS#7 a PEM

Utilizzare il comando OpenSSL pkcs7, come nell'esempio seguente. Nel seguente comando di esempio, sostituire *CertificateBundle.p7b* con il nome del file che contiene il bundle di certificati con codifica PKCS#7. Sostituire *CertificateBundle.pem* con il nome preferito del file di output per contenere il bundle di certificati con codifica PEM.

```
openssl pkcs7 -in CertificateBundle.p7b -print_certs -out CertificateBundle.pem
```

Gruppi di utenti IAM

Un gruppo di utenti IAM è una raccolta di utenti IAM. I gruppi di utenti consentono di specificare le autorizzazioni per più utenti e quindi la gestione delle autorizzazioni per quegli utenti può essere

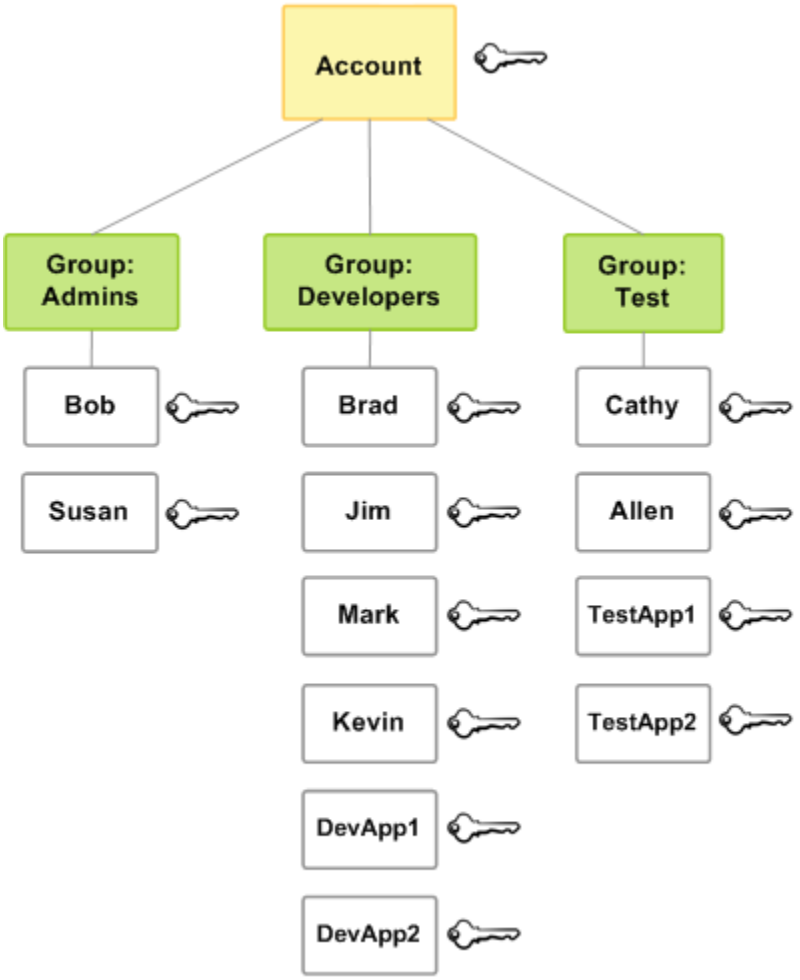
più facile. Ad esempio, potresti avere un gruppo di utenti chiamato Amministratori e concedere a tale gruppo di utenti le autorizzazioni tipiche degli amministratori. Qualsiasi utente all'interno di tale gruppo dispone automaticamente delle autorizzazioni del gruppo Amministratori. Se un nuovo utente entra a far parte dell'organizzazione e necessita dei privilegi di amministratore, puoi concedere le autorizzazioni appropriate aggiungendo l'utente al gruppo di utenti Amministratori. Se una persona cambia mansione all'interno dell'organizzazione, invece di modificare le autorizzazioni dell'utente puoi rimuoverlo dai gruppi di utenti attuali e aggiungerlo a nuovi gruppi di utenti appropriati.

Puoi collegare una policy basata sull'identità a un gruppo di utenti in modo che tutti gli utenti del gruppo ricevano le autorizzazioni della policy. Non è possibile identificare un gruppo di utenti come `Principal` in una policy (ad esempio una policy basata sulle risorse) perché i gruppi si riferiscono alle autorizzazioni, non all'autenticazione, e i principali sono entità IAM autenticate. Per ulteriori informazioni sui tipi di policy, consulta Policy basate sulle identità e policy basate su risorse.

Queste sono alcune delle caratteristiche importanti dei gruppi di utenti:

- Un gruppo di utenti può contenere molti utenti e un utente può appartenere a più gruppi di utenti.
- I gruppi di utenti non possono essere nidificati; possono contenere solo utenti, non altri gruppi di utenti.
- Non esiste alcun gruppo di utenti predefinito che include automaticamente tutti gli utenti nell' Account AWS. Se desideri disporre di un gruppo di utenti di questo tipo, è necessario crearlo e assegnarvi ogni nuovo utente.
- Il numero e la dimensione delle risorse IAM in un Account AWS, ad esempio il numero di gruppi e il numero di gruppi di cui un utente può essere membro, sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Il diagramma seguente mostra un semplice esempio di una piccola azienda. Il proprietario dell'azienda crea un gruppo di utenti `Admins` perché gli utenti possano creare e gestire altri utenti mentre l'azienda cresce. Il gruppo di utenti `Admins` crea un gruppo di utenti `Developerse` un gruppo di utenti `Test`. Ciascuno di questi gruppi di utenti è composto da utenti (umani e applicazioni) che interagiscono con AWS (Jim, Brad, DevApp 1 e così via). Ogni utente dispone di un singolo set di credenziali di sicurezza. In questo esempio, ogni utente appartiene a un singolo gruppo. Tuttavia, gli utenti possono appartenere a più gruppi di utenti.



Creazione di gruppi di utenti IAM

> **Note**
>
> Come procedura consigliata, si consiglia di richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se segui le best practice, non gestisci utenti e gruppi IAM. Al contrario, gli utenti e i gruppi sono gestiti all'esterno AWS e possono accedere alle AWS risorse come identità federata. Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web, il AWS Directory Service, la directory Identity Center o qualsiasi utente che accede AWS ai servizi utilizzando le credenziali fornite tramite un'origine di identità. Le identità federate utilizzano i gruppi definiti dal rispettivo gestore di identità. Se lo utilizzi AWS IAM Identity Center, consulta Gestisci le identità in IAM Identity Center nella Guida per l'AWS

IAM Identity Center utente per informazioni sulla creazione di utenti e gruppi in IAM Identity Center.

Per configurare un gruppo di utenti, è necessario creare il gruppo. Offrire al gruppo le autorizzazioni in base al tipo di lavoro che si prevede venga eseguito dagli utenti del gruppo. Infine, aggiungere utenti al gruppo.

Per informazioni sulle autorizzazioni di cui hai bisogno per creare un gruppo di utenti, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Come creare un gruppo di utenti IAM e collegare le policy (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione seleziona Gruppi di utenti, quindi Crea gruppo.
3. In Nome gruppo di utenti, digita il nome del gruppo.

   > **Note**
   >
   > Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote. I nomi dei gruppi possono essere una combinazione di un massimo di 128 lettere, cifre e i seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), trattino basso (_) e trattino (-). I nomi devono essere univoci nell'account. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare gruppi denominati sia **`ADMINS`** che **`admins`**.
4. Nell'elenco degli utenti, seleziona la casella di controllo per ogni utente che desideri aggiungere al gruppo.
5. Nell'elenco di tutte le policy, selezionare la casella di controllo per ogni policy che si desidera applicare a tutti i membri del gruppo.
6. Seleziona Crea gruppo.

Per creare gruppi di utenti IAM (AWS CLI o AWS API)

Utilizzare una delle seguenti operazioni:

- AWS CLI: aws iam create-group

- AWS API: CreateGroup

Gestione dei gruppi di utenti IAM

Amazon Web Services offre diversi strumenti per la gestione dei gruppi di utenti IAM. Per informazioni sulle autorizzazioni necessarie per aggiungere e rimuovere utenti in un gruppo di utenti, consulta Autorizzazioni necessarie per accedere alle risorse IAM.

Argomenti

- Visualizzazione dei gruppi di utenti IAM
- Aggiunta e rimozione di utenti in un gruppo di utenti IAM
- Collegamento di una policy a un gruppo di utenti IAM
- Ridenominazione di un gruppo di utenti IAM
- Eliminazione di un gruppo di utenti IAM

Visualizzazione dei gruppi di utenti IAM

Puoi elencare tutti i gruppi di utenti nel tuo account, elencare gli utenti in un gruppo di utenti ed elencare i gruppi di utenti a cui un utente appartiene. Se utilizzi l' AWS API AWS CLI or, puoi elencare tutti i gruppi di utenti con un particolare prefisso di percorso.

Come elencare i gruppi di utenti nel tuo account

Effettua una delle seguenti operazioni:

- AWS Management Console: nel pannello di navigazione, scegli Gruppi di utenti.
- AWS CLI: aws iam list-groups
- AWS API: ListGroups

Come elencare gli utenti in un determinato gruppo di utenti

Effettua una delle seguenti operazioni:

- AWS Management Console: nel pannello di navigazione, seleziona Gruppi di utenti, scegli il nome del gruppo e seleziona la scheda Utenti.
- AWS CLI: aws iam get-group

- AWS API: GetGroup

Come elencare tutti i gruppi di utenti in cui si trova un utente

Effettua una delle seguenti operazioni:

- AWS Management Console: nel riquadro di navigazione, scegliere Users (Utenti), il nome utente e selezionare la scheda Gruppi.
- AWS CLI: Aws Iam list-groups-for-user
- AWS API: ListGroupsForUser

Aggiunta e rimozione di utenti in un gruppo di utenti IAM

Usa i gruppi di utenti per applicare le stesse policy di autorizzazione a più utenti contemporaneamente. Potrai quindi aggiungere o rimuovere utenti da un gruppo di utenti IAM. Questa funzione è utile quando le persone arrivano e lasciano l'organizzazione.

Visualizzazione dell'accesso alle policy

Prima di modificare le autorizzazioni per una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Aggiunta o rimozione di un utente da un gruppo di utenti (console)

È possibile utilizzare il AWS Management Console per aggiungere o rimuovere un utente da un gruppo di utenti.

Come aggiungere un utente a un gruppo di utenti IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Seleziona la scheda Utenti, quindi scegli Aggiungi utenti. Selezionare la casella di controllo accanto agli utenti che si desidera aggiungere.

4. Scegli Aggiungi utenti.

Come rimuovere un utente da un gruppo IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Scegli la scheda Users (Utenti); Seleziona la casella di controllo accanto agli utenti che desideri rimuovere e quindi scegli Rimuovi utenti.

Aggiunta o rimozione di un utente da un gruppo di utenti (AWS CLI)

Puoi utilizzare il AWS CLI per aggiungere o rimuovere un utente da un gruppo di utenti.

Come aggiungere un utente a un gruppo di utenti IAM (AWS CLI)

- Utilizza il seguente comando:
  - era io add-user-to-group

Per rimuovere un utente da un gruppo di utenti IAM (AWS CLI)

- Utilizza il seguente comando:
  - era io remove-user-from-group

Aggiungere o rimuovere un utente in un gruppo di utenti (AWS API)

Puoi utilizzare l' AWS API per aggiungere o rimuovere un utente in un gruppo di utenti.

Per aggiungere un utente a un gruppo IAM (AWS API)

- Completare la seguente operazione:
  - AddUserToGroup

Per rimuovere un utente da un gruppo di utenti IAM (AWS API)

- Completare la seguente operazione:

- RemoveUserFromGroup

Collegamento di una policy a un gruppo di utenti IAM

È possibile allegare una policy AWS gestita, ovvero una policy prescritta fornita da, AWS a un gruppo di utenti, come spiegato nei passaggi seguenti. Per collegare una policy gestita dal cliente, ovvero una policy con autorizzazioni personalizzate da te creata, è prima necessario creare la policy. Per ulteriori informazioni sulla creazione di policy gestite dal cliente, consulta Creazione di policy IAM.

Per ulteriori informazioni sulle autorizzazioni e sulle policy, consulta Gestione degli accessi AWS alle risorse.

Come collegare una policy a un gruppo di utenti (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Gruppi di utenti, quindi scegli il nome del gruppo.
3. Scegli la scheda Autorizzazioni.
4. Scegli Aggiungi autorizzazioni, quindi scegli Allega politiche.
5. Le policy correnti collegate al gruppo di utenti vengono visualizzate nell'elenco Policy di autorizzazione correnti. Nell'elenco Altre policy di autorizzazioni, seleziona la casella di controllo accanto al nome delle policy da collegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy in base al tipo e al nome della policy.
6. Seleziona la policy che desideri allegare al tuo gruppo di utenti IAM e scegli Allega policy.

Per allegare una policy a un gruppo di utenti (AWS CLI o AWS API)

Esegui una delle operazioni seguenti:

- AWS CLI: era io attach-group-policy
- AWS API: AttachGroupPolicy

Ridenominazione di un gruppo di utenti IAM

Quando modifichi il nome o il percorso di un gruppo di utenti, si verificano i seguenti eventi:

- Qualsiasi policy associata al gruppo di utenti resta con il gruppo con il nuovo nome.
- Il gruppo di utenti mantiene tutti i suoi utenti con il nuovo nome.
- L'ID univoco del gruppo di utenti rimane invariato. Per ulteriori informazioni sugli ID univoci, consulta Identificatori univoci.

IAM non aggiorna automaticamente le policy che fanno riferimento al gruppo di utenti come risorsa per l'utilizzo del nuovo nome. Pertanto, è necessario prestare attenzione quando si rinomina un gruppo di utenti. Prima di rinominare il gruppo di utenti, è necessario verificare manualmente tutte le policy per individuare quelle in cui tale gruppo viene menzionato in base al nome. Supponiamo ad esempio che Bob sia il responsabile dell'area test dell'organizzazione. Bob dispone di una policy associata alla sua entità utente IAM che gli permette di aggiungere e rimuovere utenti dal gruppo di utenti Test. Se un amministratore cambia il nome del gruppo di utenti (o modifica il percorso del gruppo), deve anche aggiornare la policy associata a Bob per l'utilizzo del nuovo nome o percorso. In caso contrario Bob non potrà aggiungere o rimuovere gli utenti dal gruppo di utenti.

Per trovare le policy che fanno riferimento a un gruppo di utenti come risorsa:

1. Nel pannello di navigazione della console IAM, scegli Policy.
2. Ordina in base alla colonna Tipo per individuare le tue policy personalizzate gestite dal cliente.
3. Scegli il nome della policy da modificare.
4. Scegli la scheda Autorizzazioni e quindi Riepilogo.
5. Seleziona IAM dall'elenco di servizi, se disponibile.
6. Cerca il nome del gruppo di utenti nella colonna Risorsa.
7. Seleziona Modifica per modificare il nome del gruppo di utenti nella policy.

Come modificare il nome di un gruppo di utenti IAM

Effettua una delle seguenti operazioni:

- AWS Management Console: nel pannello di navigazione, seleziona Gruppi di utenti, quindi seleziona il nome del gruppo. Scegli Modifica. Digita il nuovo nome del gruppo di utenti e scegli Salva modifiche.
- AWS CLI: aws iam update-group
- AWS API: UpdateGroup

Eliminazione di un gruppo di utenti IAM

Quando si elimina un gruppo di utenti in AWS Management Console, la console rimuove automaticamente tutti i membri del gruppo, scollega tutte le policy gestite allegate ed elimina tutte le politiche in linea. Tuttavia, perché IAM non elimina automaticamente le policy che fanno riferimento al gruppo di utenti come risorsa, è necessario prestare attenzione quando si elimina un gruppo di utenti. Prima di poter eliminare il gruppo di utenti, devi verificare manualmente tutte le policy per individuare quelle in cui tale gruppo viene menzionato per nome. Ad esempio, John, il responsabile del team di test, dispone di una policy collegata alla sua entità utente IAM che gli consente di aggiungere e rimuovere utenti dal gruppo Test. Se un amministratore elimina il gruppo, deve eliminare anche la policy collegata a John. Altrimenti, se l'amministratore ricrea il gruppo eliminato e gli assegna lo stesso nome, le autorizzazioni di John rimangono valide, anche se ha lasciato il team di test.

Come trovare le policy che fanno riferimento a un gruppo di utenti come risorsa

1. Nel pannello di navigazione della console IAM, scegli Policy.
2. Ordina in base alla colonna Tipo per individuare le tue policy personalizzate gestite dal cliente.
3. Scegli il nome della policy da eliminare.
4. Scegli la scheda Autorizzazioni e quindi Riepilogo.
5. Seleziona IAM dall'elenco di servizi, se disponibile.
6. Cerca il nome del gruppo di utenti nella colonna Risorsa.
7. Seleziona Elimina per eliminare la policy.
8. Digita il nome della politica per confermarne l'eliminazione, quindi scegli Elimina.

Al contrario, quando si utilizza Tools for Windows PowerShell o l' AWS CLI AWS API per eliminare un gruppo di utenti, è necessario prima rimuovere gli utenti del gruppo. Quindi, elimina le policy in linea eventualmente integrate nel gruppo di utenti. Quindi, è necessario scollegare le policy gestite collegate al gruppo. Solo dopo potrai eliminare il gruppo di utenti.

Eliminazione di un gruppo di utenti IAM (console)

Puoi eliminare un gruppo di utenti IAM dalla AWS Management Console.

Come eliminare un gruppo di utenti IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione, seleziona Gruppi di utenti.
3. Nell'elenco dei gruppi di utenti, seleziona la casella di controllo accanto ai nomi dei gruppi di utenti da eliminare. È possibile utilizzare la casella di ricerca per filtrare l'elenco dei gruppi di utenti in base al tipo, alle autorizzazioni e al nome del gruppo di utenti.
4. Scegli Elimina.
5. Nella casella di conferma, se desideri eliminare un singolo gruppo di utenti, digita il nome del gruppo di utenti e scegli Elimina. Se desideri eliminare più gruppi di utenti, digita il numero di gruppi di utenti da eliminare seguito da **`user groups`** e scegli Elimina. Ad esempio, se elimini tre gruppi di utenti, digita **`3 user groups`**.

Eliminazione di un gruppo di utenti IAM (AWS CLI)

Puoi eliminare un gruppo di utenti IAM dalla AWS CLI.

Come eliminare un gruppo di utenti IAM (AWS CLI)

1. Rimuovi tutti gli utenti dal gruppo di utenti.
   - aws iam get-group (per ottenere l'elenco degli utenti nel gruppo di utenti) e aws iam remove-user-from-group (per rimuovere un utente dal gruppo di utenti)
2. Elimina tutte le policy in linea integrate nel gruppo di utenti.
   - aws iam list-group-policies (per ottenere un elenco delle politiche in linea del gruppo di utenti) e aws iam delete-group-policy (per eliminare le politiche in linea del gruppo di utenti)
3. Scollega tutte le policy gestite collegate al gruppo di utenti.
   - aws iam list-attached-group-policies (per ottenere un elenco delle politiche gestite allegate al gruppo di utenti) e aws iam detach-group-policy (per scollegare una politica gestita dal gruppo di utenti)
4. Elimina il gruppo di utenti.
   - aws iam delete-group

Eliminazione di un gruppo di utenti IAM (API)AWS

Puoi utilizzare l' AWS API per eliminare un gruppo di utenti IAM.

Per eliminare un gruppo di utenti IAM (AWS API)

1. Rimuovi tutti gli utenti dal gruppo di utenti.
   - GetGroup(per ottenere l'elenco degli utenti nel gruppo di utenti) e RemoveUserFromGroup(per rimuovere un utente dal gruppo di utenti)
2. Elimina tutte le policy in linea integrate nel gruppo di utenti.
   - ListGroupPolicies(per ottenere un elenco delle politiche in linea del gruppo di utenti) e DeleteGroupPolicy(per eliminare le politiche in linea del gruppo di utenti)
3. Scollega tutte le policy gestite collegate al gruppo di utenti.
   - ListAttachedGroupPolicies(per ottenere un elenco delle politiche gestite allegate al gruppo di utenti) e DetachGroupPolicy(per scollegare una politica gestita dal gruppo di utenti)
4. Elimina il gruppo di utenti.
   - DeleteGroup

Ruoli IAM

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. Un ruolo IAM è simile a un utente IAM, in quanto è un' AWS identità con policy di autorizzazione che determinano ciò che l'identità può e non può fare AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo.

Puoi utilizzare i ruoli per delegare l'accesso a utenti, applicazioni o servizi che normalmente non hanno accesso alle tue AWS risorse. Ad esempio, potresti voler concedere agli utenti del tuo AWS account l'accesso a risorse che di solito non dispongono o concedere agli utenti di un account Account AWS l'accesso alle risorse di un altro account. Oppure potresti voler consentire a un'app mobile di utilizzare AWS le risorse, ma non incorporare AWS le chiavi all'interno dell'app (dove possono essere difficili da aggiornare e dove gli utenti possono potenzialmente estrarle). A volte si desidera AWS consentire l'accesso a utenti che hanno già identità definite all'esterno AWS, ad esempio nella directory aziendale. In alternativa, è possibile concedere l'accesso all'account a terze parti in modo che possano eseguire un controllo sulle proprie risorse.

Per questi scenari, puoi delegare l'accesso alle AWS risorse utilizzando un ruolo IAM. Questa sezione introduce i ruoli e i diversi modi in cui è possibile utilizzarli, quando e come selezionare gli approcci, come creare, gestire, cambiare (o assumere) ed eliminare i ruoli.

> **Note**
>
> Quando crei il tuo per la prima volta Account AWS, per impostazione predefinita non viene creato alcun ruolo. Man mano che aggiungi servizi al tuo account, questi possono aggiungere ruoli collegati ai servizi per supportarne i casi d'uso.
> Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
> Prima di poter eliminare i ruoli collegati ai servizi, devi eliminare le risorse associate. Questa procedura protegge le risorse di perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.
> Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Argomenti

Termini e concetti dei ruoli

Di seguito sono elencati alcuni termini di base per aiutarti a iniziare a utilizzare i ruoli.

Ruolo

UN'identità IAM che puoi creare nell'account che ha le autorizzazioni specifiche. Un ruolo IAM presenta alcune analogie con un utente IAM. Ruoli e utenti sono entrambi identità AWS con policy di autorizzazioni che determinano ciò che l'identità può o non può fare in AWS. Tuttavia, invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque. Inoltre, un ruolo non ha credenziali a lungo termine standard associate (password o chiavi di accesso). Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo.

I ruoli possono essere utilizzati da:

- Un utente IAM con Account AWS lo stesso ruolo
- Un utente IAM con un ruolo Account AWS diverso dal ruolo
- Un servizio Web offerto da AWS Amazon Elastic Compute Cloud (Amazon EC2)
- Un utente esterno autenticato da un fornitore di servizi di identità (IdP) compatibile con SAML 2.0 o OpenID Connect o un gestore identità creato appositamente.

AWS ruolo del servizio

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

AWS ruolo di servizio per un'istanza EC2

Un tipo speciale di ruolo di servizio che un'applicazione in esecuzione su un'istanza Amazon EC2 può assumere per eseguire operazioni nell'account. Questo ruolo è assegnato all'istanza EC2 quando viene avviata. Le applicazioni in esecuzione su quell'istanza possono recuperare credenziali di sicurezza provvisorie ed eseguire le operazioni consentite dal ruolo. Per ulteriori informazioni sull'utilizzo di un ruolo di servizio per un'istanza EC2, consulta la pagina Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

AWS ruolo collegato al servizio

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

> **Note**
>
> Se stai già utilizzando un servizio quando inizia a supportare i ruoli collegati al servizio, potresti ricevere un'e-mail che annuncia un nuovo ruolo nel tuo account. In questo caso, il servizio ha creato automaticamente il ruolo collegato al servizio nel tuo account. Non è necessario compiere alcuna operazione per supportare questo ruolo e non è necessario eliminarlo manualmente. Per ulteriori informazioni, consulta Un nuovo ruolo appare nell'account AWS.

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi.

Concatenazione del ruolo

Il concatenamento dei ruoli si verifica quando si utilizza un ruolo per assumere un secondo ruolo tramite l' AWS CLI API o. Ad esempio, `RoleA` dispone dell'autorizzazione per assumere il ruolo `RoleB`. È possibile consentire a User1 di assumere `RoleA` utilizzando le proprie credenziali utente a lungo termine nell' AssumeRole operazione API. Questa restituisce le credenziali a breve termine del ruolo `RoleA`. Con la concatenazione del ruolo, puoi utilizzare le credenziali a breve termine del ruolo `RoleA` per abilitare l'Utente1 ad assumere il ruolo `RoleB`.

Quando assumi un ruolo, puoi passare un tag di sessione e impostare il tag come transitivo. I tag di sessione transitivi vengono passati a tutte le sessioni successive in una concatenazione del ruolo. Per ulteriori informazioni sui tag di sessione, consulta Passare i tag di sessione AWS STS.

Il concatenamento dei ruoli limita la sessione di ruolo AWS dell'utente AWS CLI o dell'API a un massimo di un'ora. Quando si utilizza l'operazione AssumeRoleAPI per assumere un ruolo, è possibile specificare la durata della sessione di ruolo con il `DurationSeconds` parametro. Puoi specificare un valore di parametro fino a 43200 secondi (12 ore), che dipende dall'impostazione della durata massima della sessione per il tuo ruolo. Tuttavia, se assumi un ruolo utilizzando la concatenazione dei ruoli e fornisci un valore del parametro `DurationSeconds` maggiore di un'ora, l'operazione ha esito negativo.

AWS non considera l'utilizzo di ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2 come concatenamento di ruoli.

Delega

La concessione delle autorizzazioni a un altro utente per permettere l'accesso alle risorse di controllo. La delega comporta la configurazione di un trust tra due account. Il primo è l'account proprietario della risorsa (l'account che concede fiducia). Il secondo è l'account che contiene gli utenti che devono accedere alla risorsa (l'account attendibile). L'account a cui viene concessa fiducia e l'account che concede fiducia possono essere uno dei seguenti:

- Lo stesso account.
- Account diversi che sono comunque sotto il controllo della tua organizzazione.
- Due account di proprietà di organizzazioni diverse.

Per delegare l'autorizzazione per accedere a una risorsa, crea un ruolo IAM nell'account che concede fiducia che ha due policy collegate. Le policy di autorizzazioni concedono all'utente del ruolo le autorizzazioni necessarie per eseguire le attività previste sulla risorsa. La policy di attendibilità specifica quali membri degli account a cui viene concessa fiducia sono autorizzati ad assumere il ruolo.

Quando si crea una politica di affidabilità, non è possibile specificare un carattere jolly (*) come parte di un ARN come elemento principale. La policy di affidabilità è associata al ruolo nell'account che concede fiducia e rappresenta una metà delle autorizzazioni. L'altra metà è una policy delle autorizzazioni collegata all'utente nell'account a cui viene concessa fiducia che consente a quell'utente di passare al ruolo o di assumerlo. Un utente che assume un ruolo temporaneamente cede le proprie autorizzazioni e ottiene le autorizzazioni del ruolo. Quando l'utente esce o termina l'utilizzo del ruolo, le autorizzazioni originali dell'utente vengono ripristinate. Un parametro aggiuntivo chiamato external ID contribuisce a garantire sicuro l'uso dei ruoli tra gli account che non vengono controllati dalla stessa organizzazione.

Federazione

La creazione di una relazione di fiducia tra un provider di identità esterno e. AWSGli utenti possono accedere a un provider OIDC, ad esempio Login with Amazon, Facebook, Google o qualsiasi IdP compatibile con OpenID Connect (OIDC). Gli utenti possono anche effettuare l'accesso a un sistema di identità enterprise compatibile con Security Assertion Markup Language (SAML) 2.0, ad esempio Microsoft Active Directory Federation Services. Quando utilizzi OIDC e SAML 2.0 per configurare una relazione di fiducia tra questi provider di identità esterni e AWS, all'utente viene assegnato un ruolo IAM. L'utente riceve anche credenziali temporanee che gli consentono di accedere alle risorse. AWS

Utente federato

Invece di creare un utente IAM, puoi utilizzare le identità esistenti provenienti dalla AWS Directory Service tua directory utente aziendale o da un provider OIDC. Questi sono noti come utenti federati. AWS assegna un ruolo a un utente federato quando l'accesso viene richiesto tramite un provider di identità. Per ulteriori informazioni sugli utenti federati, consulta Utenti federati e ruoli.

Policy di trust

Documento di policy JSON in cui si definiscono i principali considerati attendibili per assumere il ruolo. Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo in IAM. I principali che è possibile specificare nella policy di attendibilità includono utenti, ruoli, account e servizi.

Policy delle autorizzazioni

Un documento delle autorizzazioni in formato JSON in cui definisci le operazioni e le risorse che il ruolo può utilizzare. Il documento è scritto in base alle regole del linguaggio della policy IAM.

Limite delle autorizzazioni

Una funzione avanzata in cui usi le policy per limitare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un ruolo. Non è possibile applicare un limite delle autorizzazioni a un ruolo collegato al servizio. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM.

Principale

Un'entità in AWS grado di eseguire azioni e accedere alle risorse. Un principale può essere un Utente root dell'account AWS utente IAM o un ruolo. Puoi concedere le autorizzazioni ad accedere a una risorsa in uno dei seguenti due modi:

- Puoi collegare una policy delle autorizzazioni a un utente (direttamente o indirettamente tramite un gruppo) o a un ruolo.
- Per quei servizi che supportano le resource-based policies, puoi identificare la principale nell'elemento `Principal` di una policy collegata alla risorsa.

Se si fa riferimento a un Account AWS come principale, in genere si intende qualsiasi principale definito all'interno di quell'account.

> **Note**
>
> Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN in una politica di affidabilità del ruolo. Per informazioni dettagliate, vedi AWS Elementi della policy JSON: Principal.

Ruolo per l'accesso tra account

Un ruolo che concede l'accesso alle risorse in un account a un principale affidabile in un diverso account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, alcuni dei servizi AWS ti consentono di collegare una policy direttamente a una risorsa (invece di utilizzare un ruolo come proxy). Queste sono chiamate politiche basate sulle risorse ed è possibile utilizzarle per concedere ai responsabili di un'altra persona l' Account AWS accesso alla risorsa. Alcune di queste risorse includono bucket Amazon Simple Storage Service (S3), vault S3 Glacier, argomenti Amazon Simple Notification Service (SNS) e code Amazon Simple Queue Service (SQS). Per informazioni su quali servizi supportano le policy basate su risorse, consulta AWS servizi che funzionano con IAM. Per ulteriori informazioni sulle policy basate sulle risorse, consulta Accesso alle risorse multi-account in IAM.

Scenari comuni per ruoli: utenti, applicazioni e servizi

Come per la maggior parte delle AWS funzionalità, in genere hai due modi per utilizzare un ruolo: in modo interattivo nella console IAM o a livello di codice con gli AWS CLI strumenti per Windows PowerShell o l'API.

- Gli utenti IAM nell'account che utilizza la console IAM possono passare a un ruolo per utilizzare temporaneamente le autorizzazioni del ruolo nella console. Gli utenti abbandonano le loro autorizzazioni originali e assumono le autorizzazioni assegnate al ruolo. Quando gli utenti escono dal ruolo, le autorizzazioni originali vengono ripristinate.
- Un'applicazione o un servizio offerto da AWS (come Amazon EC2) può assumere un ruolo richiedendo credenziali di sicurezza temporanee per un ruolo a cui effettuare richieste programmatiche. AWSÈ possibile utilizzare un ruolo in questo modo per non dover condividere o gestire le credenziali di sicurezza a lungo termine (ad esempio creando un utente IAM) per ogni entità che richiede l'accesso a una risorsa.

> **Note**
>
> In questa guida le frasi passare a un ruolo e assumere un ruolo vengono utilizzate in modo intercambiabile.

Il modo più semplice per utilizzare i ruoli è quello di concedere agli utenti IAM le autorizzazioni per passare ai ruoli creati da te all'interno del tuo o di un altro Account AWS. È possibile passare da un ruolo all'altro facilmente utilizzando la console IAM per utilizzare le autorizzazioni che non si desidera abbiano normalmente e uscire dal ruolo per cedere a tali autorizzazioni. Ciò può aiutare a impedire l'accesso accidentale alle risorse sensibili o la loro modifica.

Per utilizzi più complessi di ruoli, ad esempio la concessione di accesso alle applicazioni e servizi, o gli utenti federati esterni, è possibile richiamare l'API `AssumeRole`. Questa chiamata API restituisce un set di credenziali temporanee che l'applicazione può utilizzare in successive chiamate API. Le operazioni tentate con le credenziali temporanee dispongono solo delle autorizzazioni concesse dal ruolo associato. Un'applicazione non deve "uscire" dal ruolo nello stesso modo di un utente nella console, ma l'applicazione smette semplicemente di utilizzare le credenziali temporanee e riprende le chiamate con le credenziali originali.

Gli utenti federati accedono utilizzando le credenziali di un provider di identità (IdP). AWS fornisce quindi credenziali temporanee all'IdP affidabile da trasmettere all'utente per includerle nelle AWS successive richieste di risorse. Queste credenziali forniscono le autorizzazioni concesse al ruolo assegnato.

Questa sezione fornisce una panoramica dei seguenti scenari:

- Fornisci l'accesso a un utente IAM in uno Account AWS di tua proprietà per accedere alle risorse di un altro account di tua proprietà
- Fornire l'accesso a carichi di lavoro non AWS
- Fornire l'accesso agli utenti IAM negli Account AWS di proprietà di terze parti
- Fornisci l'accesso ai servizi offerti dalle AWSAWS risorse
- Fornire l'accesso agli utenti autenticati esternamente (federazione delle identità)

Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà

Puoi concedere ai tuoi utenti IAM il permesso di passare a ruoli interni a te Account AWS o a ruoli definiti in altri ruoli Account AWS di tua proprietà.

> **Note**
>
> Se desideri concedere l'accesso a un account che non possiedi né controlli, consulta Fornire l'accesso a Account AWS siti di proprietà di terzi più avanti in questo argomento.

Immaginiamo di avere delle istanze Amazon EC2 critiche per la tua organizzazione. Invece di concedere direttamente agli utenti l'autorizzazione a terminare le istanze, è possibile creare un ruolo con tali privilegi. Quindi consentire agli amministratori di passare al ruolo quando è necessario terminare un'istanza. In questo modo si aggiungono i seguenti livelli di protezione alle istanze:

- È necessario concedere esplicitamente agli utenti il permesso di assumere quel ruolo.
- I tuoi utenti devono passare attivamente al ruolo utilizzando AWS Management Console o assumere il ruolo utilizzando l' AWS API AWS CLI o.
- È possibile aggiungere una Multi-Factor Authentication (MFA) al ruolo, in modo che solo gli utenti che accedono con un dispositivo MFA possano assumere quel ruolo. Per ulteriori informazioni su come configurare un ruolo in modo che gli utenti che assumono il ruolo debbano essere prima autenticati utilizzando l'autenticazione a più fattori (MFA), consulta Configurazione dell'accesso alle API protetto da MFA.

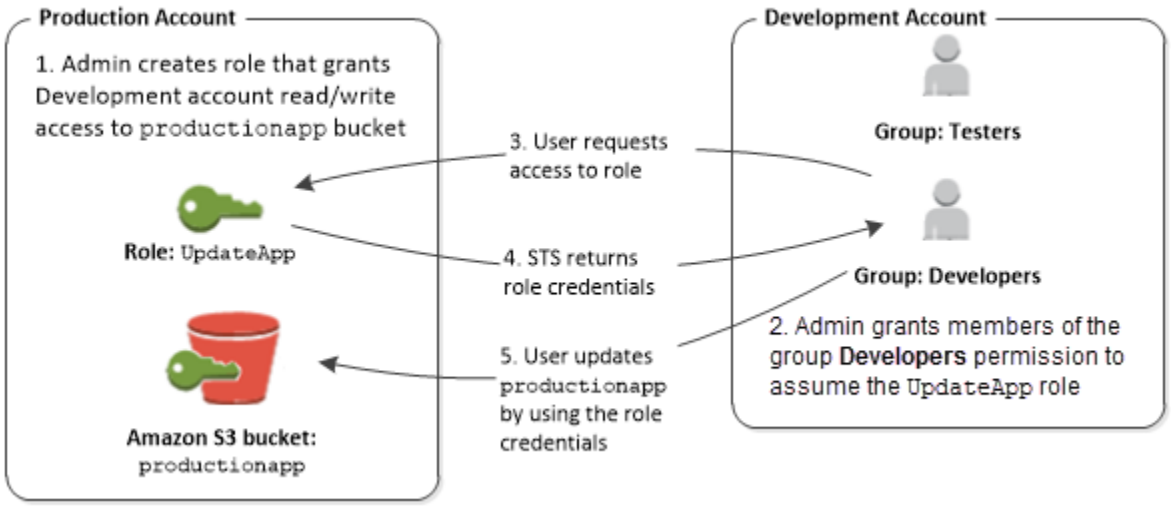
Consigliamo di utilizzare questo approccio per applicare il principio di privilegio minimo. Ciò significa limitare l'uso di autorizzazioni elevate unicamente a quelle volte in cui sono necessarie per operazioni specifiche. Per impedire le modifiche accidentali apportate agli ambienti sensibili, puoi utilizzare i ruoli, soprattutto se combinati con attività di audit per garantire che vengano utilizzati solo quando necessario.

Quando si crea un ruolo per questo scopo, è necessario specificare l'ID degli account da cui gli utenti devono accedere nell'elemento `Principal` della policy di affidabilità del ruolo. È quindi possibile concedere agli utenti specifici in tali altri account le autorizzazioni per passare al ruolo. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Un utente in un account può passare a un ruolo dello stesso o di un altro account. Mentre si usa il ruolo, l'utente è in grado di eseguire solo le azioni e accedere solo alle risorse consentite dal ruolo; le loro autorizzazioni utente originali sono sospese. Quando l'utente esce dal ruolo, le autorizzazioni utente originali vengono ripristinate.

Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati

Immaginate che la vostra organizzazione disponga Account AWS di più elementi per isolare un ambiente di sviluppo da un ambiente di produzione. Gli utenti nell'account di sviluppo potrebbero occasionalmente aver bisogno di accedere alle risorse nell'account di produzione. Ad esempio, potrebbe essere necessario l'accesso a più account quando si sta richiedendo un aggiornamento dall'ambiente di sviluppo all'ambiente di produzione. Anche se è possibile creare identità separate (e password) per gli utenti che lavorano con entrambi gli account, la gestione delle credenziali per più account complica la gestione delle identità. Nell'illustrazione seguente, tutti gli utenti vengono gestiti nell'account di sviluppo, ma per alcuni sviluppatori è necessario un accesso limitato all'account di produzione. L'account di sviluppo dispone di due gruppi: collaudatori e sviluppatori, e ciascun gruppo ha la propria policy.



1. Nell'account di produzione, un amministratore utilizza IAM per creare il ruolo `UpdateApp` in tale account. Nel ruolo, l'amministratore definisce una policy di affidabilità che specifica l'account di sviluppo come `Principal`; in tal modo gli utenti autorizzati dall'account di sviluppo possono utilizzare il ruolo `UpdateApp`. L'amministratore definisce inoltre una policy delle autorizzazioni per il ruolo che specifica le autorizzazioni in lettura e scrittura per il bucket Amazon S3 denominato `productionapp`.

L'amministratore quindi condivide le informazioni appropriate con chiunque debba assumere quel ruolo. Tali informazioni sono il numero di account e il nome del ruolo (per gli utenti della AWS console) o l'Amazon Resource Name (ARN) (per AWS CLI l'accesso all' AWS API). L'ARN del ruolo può essere simile a `arn:aws:iam::123456789012:role/UpdateApp`, dove il ruolo è denominato `UpdateApp` ed è stato creato nel numero di account 123456789012.

> **Note**
>
> L'amministratore può eventualmente configurare il ruolo in modo che gli utenti che assumono il ruolo debbano essere prima autenticati utilizzando l'autenticazione a più fattori (MFA). Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto da MFA.

2. Nell'account di sviluppo, un amministratore concede ai membri del gruppo di sviluppatori l'autorizzazione a cambiare il ruolo. Ciò viene fatto concedendo al gruppo Developers l'autorizzazione a chiamare l'`AssumeRole`API AWS Security Token Service (AWS STS) per il `UpdateApp` ruolo. Qualsiasi utente IAM che appartiene al gruppo Sviluppatori nell'account di sviluppo può ora passare al ruolo `UpdateApp` nell'account di produzione. Gli altri utenti che non appartengono al gruppo di sviluppatori non hanno il permesso di passare al ruolo e pertanto non sono in grado di accedere al bucket S3 nell'account di produzione.
3. L'utente richiede di cambiare il ruolo:
   - AWS console: l'utente sceglie il nome dell'account nella barra di navigazione e sceglie Switch Role. L'utente specifica l'ID account (o alias) e il nome del ruolo. In alternativa, l'utente può fare clic su un collegamento inviato nell'e-mail dall'amministratore. Il link indirizza l'utente alla pagina Switch Role (Cambia ruolo) con i dettagli già compilati.
   - AWS API/AWS CLI: Un utente del gruppo Developers dell'account di sviluppo chiama la `AssumeRole` funzione per ottenere le credenziali per il ruolo. `UpdateApp` L'utente specifica l'ARN del ruolo `UpdateApp` come parte della chiamata. Se un utente nel gruppo di collaudatori inoltra la stessa richiesta, la richiesta ha esito negativo perché i collaudatori non hanno l'autorizzazione a chiamare `AssumeRole` per il `UpdateApp` ruolo ARN.
4. AWS STS restituisce credenziali temporanee:
   - AWS console: AWS STS verifica la richiesta con la politica di fiducia del ruolo per garantire che la richiesta provenga da un'entità attendibile (che è: l'account di sviluppo). Dopo la verifica, AWS STS restituisce le credenziali di sicurezza temporanee alla AWS console.

- API/CLI: AWS STS verifica la richiesta rispetto alla politica di fiducia del ruolo per garantire che la richiesta provenga da un'entità attendibile (che è: l'account Development). Dopo la verifica, AWS STS restituisce le credenziali di sicurezza temporanee all'applicazione.

5. Le credenziali temporanee consentono l'accesso alla AWS risorsa:
   - AWS console: la AWS console utilizza le credenziali temporanee per conto dell'utente per tutte le azioni successive della console, in questo caso, per leggere e scrivere nel `productionapp` bucket. La console non è in grado di accedere ad altre risorse nell'account di produzione. Quando l'utente esce dal ruolo, le autorizzazioni dell'utente tornano a quelle originali detenute prima di cambiare il ruolo.
   - API/CLI: l'applicazione utilizza le credenziali di sicurezza provvisorie per aggiornare il bucket `productionapp`. Con le credenziali di sicurezza provvisorie, l'applicazione può solo leggere e scrivere al bucket `productionapp` e non è in grado di accedere a qualsiasi altra risorsa nell'account di produzione. L'applicazione non deve uscire dal ruolo, bensì cessa di utilizzare le credenziali provvisorie e utilizza le credenziali originali nelle successive chiamate API.

Ulteriori informazioni

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM

Fornire accesso a carichi di lavoro non AWS

Un ruolo IAM è un oggetto in AWS Identity and Access Management (IAM) a cui vengono assegnate le autorizzazioni. Quando assumi quel ruolo utilizzando un'identità IAM o un'identità esterna a AWS, ti vengono fornite credenziali di sicurezza temporanee per la tua sessione di ruolo. Potresti avere carichi di lavoro in esecuzione nel tuo data center o in un'altra infrastruttura AWS che non richiedono l'accesso alle tue AWS risorse. Invece di creare, distribuire e gestire chiavi di accesso a lungo termine, puoi utilizzare AWS Identity and Access Management Roles Anywhere (IAM Roles Anywhere) per autenticare i tuoi carichi non di lavoro. AWS IAM Roles Anywhere utilizza i certificati X.509 dell'autorità di certificazione (CA) per autenticare le identità e fornire l'accesso in modo sicuro alle credenziali temporanee fornite da Servizi AWS un ruolo IAM.

Per utilizzare IAM Roles Anywhere, è necessario configurare una CA utilizzando AWS Private Certificate Authority o utilizzare una CA dalla propria infrastruttura PKI. Dopo aver impostato una CA, si crea un oggetto in IAM Roles Anywhere chiamato Trust anchor per stabilire un rapporto di fiducia tra IAM Roles Anywhere e la tua CA per l'autenticazione. Puoi quindi configurare i ruoli IAM esistenti

o creare nuovi ruoli che si fidino del servizio IAM Roles Anywhere. Quando i tuoi AWS carichi di lavoro non si autenticano con IAM Roles Anywhere utilizzando il trust anchor, possono ottenere credenziali temporanee per i tuoi ruoli IAM per accedere alle tue risorse. AWS

Per ulteriori informazioni sulla configurazione di Ruoli IAM Anywhere, consulta l'argomento Cos'è AWS Identity and Access Management Ruoli Anywhere nella Guida dell'utente di IAM Roles Anywhere.

Fornire l'accesso a Account AWS siti di proprietà di terzi

Quando terze parti richiedono l'accesso alle AWS risorse dell'organizzazione, puoi utilizzare i ruoli per delegare l'accesso a tali risorse. Ad esempio, una terza parte potrebbe fornire un servizio per la gestione delle risorse AWS . Con i ruoli IAM, puoi concedere a queste terze parti l'accesso alle tue AWS risorse senza condividere le tue credenziali AWS di sicurezza. Invece, la terza parte può accedere alle tue AWS risorse assumendo un ruolo da te creato all'interno delle tue. Account AWS Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Le terze parti devono fornirti le informazioni seguenti per permetterti di creare un ruolo che possa essere da loro assunto:

- L' Account AWS ID della terza parte. Si specifica il loro Account AWS ID come principale quando si definisce la politica di fiducia per il ruolo.
- ID esterno da associare in modo univoco con il ruolo. L'ID esterno può essere qualsiasi identificatore noto a te e alla terza parte. Puoi ad esempio usare un ID di fattura tra te e la terza parte, ma non devi usare qualcosa che sia possibile indovinare, ad esempio il nome o il numero di telefono della terza parte. Devi specificare questo ID quando definisci la policy di affidabilità per il ruolo. La terza parte deve fornire questo ID quando assume il ruolo. Per ulteriori informazioni sull'ID esterno, consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.
- Le autorizzazioni richieste dalla terza parte per utilizzare le tue AWS risorse. Devi specificare queste autorizzazioni quando definisci la policy di autorizzazione del ruolo. Questa policy definisce le operazioni consentite e le risorse a cui è possibile accedere.

Dopo aver creato il ruolo, devi fornire l'Amazon Resource Name (ARN) del ruolo alla terza parte. L'ARN del ruolo è necessario per assumere il ruolo.

> **Important**
>
> Quando concedi a terze parti l'accesso alle tue AWS risorse, queste possono accedere a qualsiasi risorsa specificata nella politica. Le risorse usate dalla terza parte vengono fatturate a te. Assicurati di limitare l'uso delle risorse in modo appropriato.

Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi

A volte, è necessario concedere a terzi l'accesso alle proprie AWS risorse (accesso delegato). Un aspetto importante di questo scenario è l'ID esterno, informazioni facoltative che è possibile utilizzare in una policy di attendibilità del ruolo IAM per indicare chi può assumere il ruolo.

> **Important**
>
> AWS non considera l'ID esterno come segreto. Dopo aver creato un segreto, ad esempio una coppia di chiavi di accesso o una password AWS, non è possibile visualizzarli nuovamente. L'ID esterno per un ruolo può essere visualizzato da tutti gli utenti che dispongono dell'autorizzazione per visualizzare il ruolo.

In un ambiente multi-tenant in cui si supportano più clienti con AWS account diversi, si consiglia di utilizzare un ID esterno per utente. Account AWS Questo ID dovrebbe essere una stringa casuale generata dalla terza parte.

Per richiedere che la terza parte fornisca un ID esterno quando si assume un ruolo, aggiorna la policy di attendibilità del ruolo con l'ID esterno scelto.

Per fornire un ID esterno quando assumi un ruolo, utilizza l' AWS API AWS CLI o per assumere quel ruolo. Per ulteriori informazioni, consulta l'operazione dell'AssumeRoleAPI STS o l'operazione CLI STS assume-role.

Ad esempio, supponiamo che decidiate di assumere una società terza chiamata Example Corp per monitorare e ottimizzare i Account AWS costi. Per tenere traccia delle spese giornaliere, Example Corp deve accedere alle tue AWS risorse. Example Corp controlla anche molti altri account AWS per altri clienti.

Non fornire l'accesso a Example Corp a un utente IAM e le relative credenziali a lungo termine nell'account AWS . Utilizza invece un ruolo IAM e le credenziali di sicurezza temporanee. Un ruolo

IAM fornisce un meccanismo per consentire a terzi di accedere alle vostre AWS risorse senza dover condividere credenziali a lungo termine (come una chiave di accesso utente IAM).

Puoi utilizzare un ruolo IAM per stabilire una relazione di fiducia tra il tuo account Account AWS e quello di Example Corp. Dopo aver stabilito questa relazione, un membro dell'account Example Corp può chiamare l' AWS Security Token Service AssumeRoleAPI per ottenere credenziali di sicurezza temporanee. I membri di Example Corp possono quindi utilizzare le credenziali per accedere alle AWS risorse del tuo account.

> **Note**
>
> Per ulteriori informazioni sulle AssumeRole e altre operazioni AWS API che è possibile chiamare per ottenere credenziali di sicurezza temporanee, vedere. Richiesta di credenziali di sicurezza temporanee

Di seguito è illustrata un'analisi più dettagliata di questo scenario.

1. L'Utente A affida un incarico a Example Corp, che crea un identificatore univoco per l'Utente A. Ti forniscono questo ID cliente univoco e il loro Account AWS numero. Queste informazioni sono necessarie per creare un ruolo IAM nella fase successiva.

   > **Note**
   >
   > Example Corp può utilizzare qualsiasi valore di stringa desiderato per il ExternalId, purché sia unico per ogni cliente. È possibile che si tratti di un numero di account cliente o addirittura di una stringa di caratteri casuale, purché non esistano due clienti con lo stesso valore. Non si tratta di un "segreto". Example Corp deve fornire il ExternalId valore a ciascun cliente. L'aspetto cruciale è che l'ID deve essere generato da Example Corp e non dai clienti affinché ogni ID esterno sia univoco.

2. Accedi AWS e crei un ruolo IAM che consente a Example Corp di accedere alle tue risorse. Come per qualsiasi ruolo IAM, il ruolo dispone di due tipi di policy: una policy di autorizzazione e una policy di attendibilità. La policy di affidabilità del ruolo specifica chi può assumere il ruolo. Nel nostro scenario di esempio, la policy specifica il Account AWS numero di Example Corp come. `Principal` Ciò consente alle identità di tale account di assumere il ruolo. Inoltre, viene aggiunto un elemento `Condition` alla policy di attendibilità. Questo elemento `Condition` verifica la chiave

di contesto `ExternalId` per assicurarsi che corrisponda all'ID cliente univoco di Example Corp. Ad esempio:

```
   "Principal": {"AWS": "Example Corp's Account AWS ID"},
   "Condition": {"StringEquals": {"sts:ExternalId": "Unique ID Assigned by Example
Corp"}}
```

3. La policy di autorizzazione per il ruolo specifica le operazioni che il ruolo consente di effettuare a un utente. Ad esempio, puoi specificare che il ruolo deve permettere agli utenti di gestire solo le risorse Amazon RDS e Amazon EC2, ma non gli utenti o i gruppi IAM. In questo scenario di esempio, si utilizza la policy di autorizzazione per fornire l'accesso in sola lettura per Example Corp a tutte le risorse nell'account dell'Utente A.
4. Dopo aver creato il ruolo, è necessario fornire l'Amazon Resource Name (ARN) del ruolo a Example Corp.
5. Quando Example Corp deve accedere alle tue AWS risorse, qualcuno dell'azienda chiama l'API. AWS `sts:AssumeRole` La chiamata include l'ARN del ruolo da assumere e il ExternalId parametro che corrisponde all'ID cliente.

Se la richiesta proviene da qualcuno che utilizza Example Corp e se l'ARN del ruolo e l'ID esterno sono corretti, la richiesta ha esito positivo. Account AWS Fornisce quindi credenziali di sicurezza temporanee che Example Corp può utilizzare per accedere alle AWS risorse consentite dal ruolo.

In altre parole, quando una policy di ruolo include un ID esterno, chiunque desideri assumere il ruolo deve essere un entità principale nel ruolo e deve includere l'ID esterno corretto.

Perché utilizzare un ID esterno?

In termini astratti, l'ID esterno consente all'utente che sta assumendo il ruolo di dichiarare le circostanze in cui sta operando. Fornisce inoltre un modo per il proprietario dell'account di consentire che il ruolo venga assunto solo in circostanze specifiche. La funzione principale dell'ID esterno è quella di risolvere e prevenire il Problema del "confused deputy".

Quando si deve usare l'ID esterno?

Utilizzare un ID esterno nelle seguenti situazioni:

- Sei un Account AWS proprietario e hai configurato un ruolo per una terza parte che accede ad altri ruoli oltre Account AWS al tuo. È opportuno chiedere alla terza parte un ID esterno da includere

quando assume il ruolo fornito alla terza parte. Quindi verificare l'ID esterno tramite la policy di affidabilità del ruolo fornito alla terza parte. Ciò garantisce che la parte esterna possa assumere il tuo ruolo solo quando agisce per conto del proprietario.

- Ci si trova in una posizione che comporta l'assunzione di ruoli per conto di diversi clienti in modo analogo a Example Corp nello scenario precedente. È opportuno assegnare un ID esterno univoco a ciascun cliente e fornire indicazioni per aggiungere l'ID esterno alla policy di affidabilità creata per il ruolo da fornire. È quindi necessario assicurarsi di includere sempre l'ID esterno corretto nelle richieste di assunzione dei ruoli.

  Probabilmente si dispone già di un identificativo univoco per ogni cliente e questo ID univoco è sufficiente per l'utilizzo come ID esterno. L'ID esterno non è un valore speciale da creare in modo esplicito o monitorare separatamente, solo per questo scopo.

  Si deve sempre specificare l'ID esterno nelle chiamate API `AssumeRole`. Inoltre, quando un cliente assegna un ARN del ruolo, verificare se è possibile assumere il ruolo con e senza l'ID esterno corretto. Se è possibile assumere il ruolo senza l'ID esterno corretto, non memorizzare l'ARN del ruolo del cliente nel sistema. Attendere fino a quando il cliente non ha aggiornato la policy di affidabilità del ruolo per richiedere l'ID esterno corretto. In questo modo è possibile aiutare i clienti a operare nel modo corretto e pertanto a garantire la sicurezza di entrambi rispetto al problema "confused deputy".

Fornire l'accesso a un servizio AWS

Molti AWS servizi richiedono l'utilizzo di ruoli per controllare a cosa può accedere quel servizio. Un ruolo che un servizio assume per eseguire operazioni a tuo nome viene chiamato ruolo del servizio. Quando un ruolo fornisce uno scopo specializzato per un servizio, può essere categorizzato come ruolo del servizio per le istanze EC2 o come ruolo collegato al servizio. Consulta la documentazione AWS di ciascun servizio per verificare se utilizza ruoli e per ulteriori informazioni su come assegnare un ruolo per il servizio da utilizzare.

Per informazioni dettagliate sulla creazione di un ruolo per delegare l'accesso a un servizio offerto da AWS, consultaCreazione di un ruolo per delegare le autorizzazioni a un servizio AWS.

Problema del "confused deputy"

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell’autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Per evitare che ciò accada, AWS fornisce strumenti che ti aiutano a

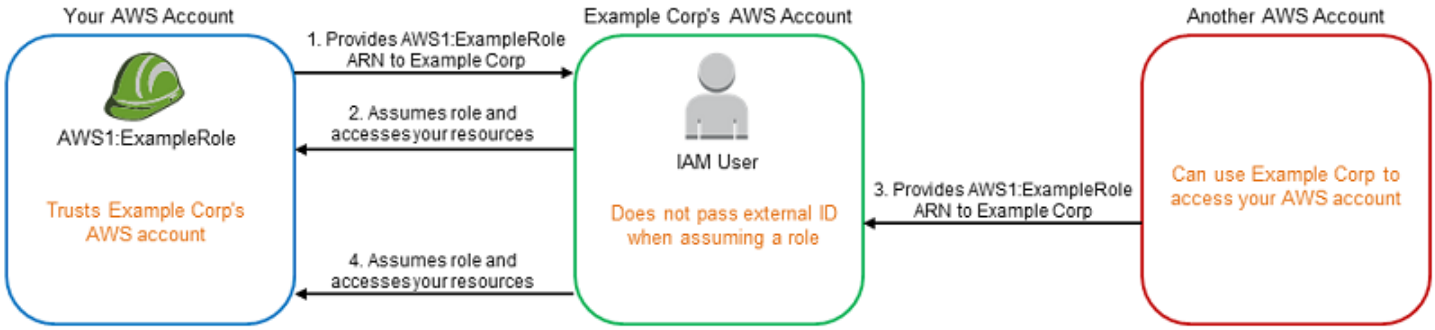
proteggere il tuo account se fornisci a terzi (i cosiddetti cross-account) o ad altri AWS servizi (noti come cross-service) l'accesso alle risorse del tuo account.

A volte, potresti dover concedere a terzi l'accesso alle tue AWS risorse (accesso delegato). Ad esempio, supponiamo che tu decida di assumere una società terza chiamata Example Corp per monitorare Account AWS e ottimizzare i costi. Per tenere traccia delle spese giornaliere, Example Corp deve accedere alle tue AWS risorse. Example Corp ne monitora anche molte altre Account AWS per conto di altri clienti. Puoi utilizzare un ruolo IAM per stabilire una relazione di fiducia tra il tuo account Account AWS e quello di Example Corp. Un aspetto importante di questo scenario è l'ID esterno, informazioni facoltative che è possibile utilizzare in una policy di attendibilità del ruolo IAM per indicare chi può assumere il ruolo. La funzione principale dell'ID esterno è quella di risolvere e prevenire il problema del "confused deputy" (delegato confuso).

Nel frattempo AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso.

Prevenzione del problema "confused deputy" tra account

Il seguente diagramma illustra il problema "confused deputy" tra account.



In questo scenario sono validi i requisiti riportati di seguito:

- AWS 1 è tuo. Account AWS
- AWS 1: ExampleRole è un ruolo nel tuo account. La policy di affidabilità di questo ruolo considera attendibile Example Corp specificando l'account AWS di Example Corp come account che può assumere il ruolo.

Ecco che cosa succede:

1. Quando inizi a utilizzare il servizio di Example Corp, fornisci l'ARN AWS di 1 ExampleRole: a Example Corp.
2. Example Corp utilizza quel ruolo ARN per ottenere credenziali di sicurezza temporanee per accedere alle risorse del tuo. Account AWS In questo modo, l'Utente A considera Example Corp come "deputy" attendibile che può agire per conto dell'Utente A stesso.
3. Anche un altro AWS cliente inizia a utilizzare il servizio di Example Corp e fornisce anche l'ARN AWS di 1 ExampleRole: for Example Corp da utilizzare. Presumibilmente l'altro cliente ha imparato o indovinato il numero AWS 1: ExampleRole, che non è un segreto.
4. Quando l'altro cliente chiede a Example Corp di accedere alle AWS risorse del suo account (quello che afferma di essere), Example Corp utilizza AWS 1: ExampleRole per accedere alle risorse del tuo account.

Questo è il modo in cui altri clienti possono ottenere l'accesso non autorizzato alle risorse di un utente, in questo caso dell'Utente A. Poiché il cliente Utente B è stato in grado di ingannare Example Corp e lo ha indotto ad agire involontariamente sulle risorse, Example Corp è ora un "confused deputy".

Example Corp può risolvere il problema "confused deputy" chiedendo di includere la condizione di verifica `ExternalId` nella policy di affidabilità del ruolo. Example Corp genera un valore `ExternalId` univoco per ogni cliente e lo utilizza nella sua richiesta per assumere il ruolo. Il valore `ExternalId` deve essere univoco tra i clienti di Example Corp e controllato da Example Corp, non dai suoi clienti. Questo è il motivo per cui i clienti lo ricevono da Example Corp e non lo creano in autonomia. In questo modo si evita che Example Corp si comporti in modo confuso e consenta l'accesso alle risorse di un altro account. AWS

In questo scenario, immagina che l'ID univoco di Example Corp per te sia 12345 e quello per l'altro cliente sia 67890. Questi ID sono semplificati per comodità in questo scenario. In genere, questi identificatori sono GUID. Supponendo che questi identificatori siano univoci tra i clienti di Example Corp, sono valori sensibili da utilizzare per l'ID esterno.

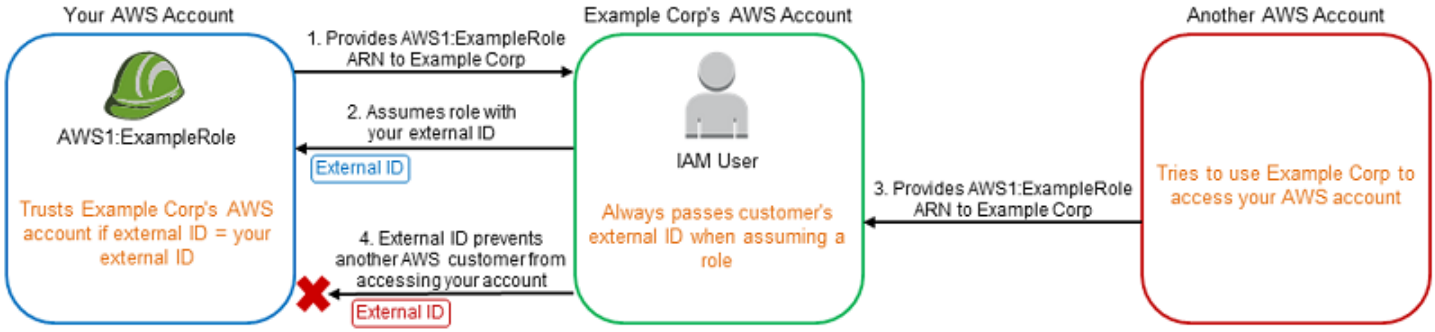
Example Corp ti fornisce il valore ID esterno 12345. È necessario aggiungere un elemento `Condition` alla policy di affidabilità del ruolo che richieda che il valore `sts:ExternalId` sia 12345, come segue:

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Principal": {
      "AWS": "Example Corp's AWS Account ID"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "12345"
      }
    }
  }
}
```

L'elemento Condition di questa politica consente a Example Corp di assumere il ruolo solo quando la chiamata AssumeRole API include il valore ID esterno 12345. Example Corp si assicura che ogni volta che assume un ruolo per conto di un cliente, includa sempre il valore dell'ID esterno del cliente nella chiamata. AssumeRole Anche se un altro cliente fornisce a Example Corp il tuo ARN, non può controllare l'ID esterno che Example Corp include nella sua richiesta. AWS In questo modo è possibile evitare che un cliente non autorizzato acceda alle tue risorse.

Il diagramma seguente illustra tale processo.



1. Come in precedenza, quando si inizia a utilizzare il servizio di Example Corp, si fornisce l'ARN AWS di 1 ExampleRole: a Example Corp.
2. Quando Example Corp utilizza quel ruolo ARN per assumere il AWS ruolo 1 ExampleRole:, Example Corp include l'ID esterno (12345) nella chiamata API. AssumeRole L'ID esterno corrisponde alla politica di fiducia del ruolo, quindi la chiamata AssumeRole API ha esito positivo e Example Corp ottiene le credenziali di sicurezza temporanee per accedere alle risorse del tuo. Account AWS
3. Anche un altro AWS cliente inizia a utilizzare il servizio di Example Corp e, come in precedenza, fornisce anche l'ARN AWS di 1 ExampleRole: for Example Corp da utilizzare.

4. Ma questa volta, quando Example Corp tenta di assumere il ruolo AWS 1: ExampleRole, fornisce l'ID esterno associato all'altro cliente (67890). L'altro cliente non ha modo di modificare questa operazione. Example Corp opera in questo modo perché la richiesta di utilizzare il ruolo proviene dall'altro cliente, pertanto 67890 indica la circostanza in cui Example Corp sta operando. Poiché hai aggiunto una condizione con il tuo ID esterno (12345) alla politica di fiducia AWS 1: ExampleRole, la AssumeRole chiamata API ha esito negativo. All'altro cliente viene impedito di ottenere l'accesso non autorizzato alle risorse nel tuo account (indicato dalla "X" rossa nel diagramma).

L'ID esterno consente di impedire a qualsiasi altro cliente di ingannare Example Corp e indurre l'azienda ad accedere involontariamente alle risorse.

Prevenzione del problema "confused deputy" tra servizi

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` o `aws:SourceOrgPaths` nelle policy basate sulle risorse per limitare le autorizzazioni di un servizio nei confronti di una risorsa specifica. `aws:SourceArn`Da utilizzare per associare una sola risorsa all'accesso tra servizi. Utilizzato `aws:SourceAccount` per consentire che qualsiasi risorsa in quell'account venga associata all'utilizzo tra servizi. Utilizzato `aws:SourceOrgID` per consentire l'associazione di qualsiasi risorsa di qualsiasi account all'interno di un'organizzazione all'utilizzo tra servizi. Da utilizzare `aws:SourceOrgPaths` per associare qualsiasi risorsa proveniente dagli account all'interno di un AWS Organizations percorso all'utilizzo tra servizi. Per ulteriori informazioni sull'utilizzo e la conoscenza dei percorsi, consulta Comprendere il percorso dell'entità AWS Organizations.

Il modo più granulare per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa nelle proprie policy basate sulle risorse. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, è necessario utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (`*`) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare sia `aws:SourceAccount` che `aws:SourceArn` per limitare le autorizzazioni.

Per proteggersi dal problema "confused deputy" su larga scala, nelle policy basate sulle risorse utilizza la chiave di contesto della condizione globale `aws:SourceOrgID` o `aws:SourceOrgPaths`

con l'ID dell'organizzazione o il percorso dell'organizzazione della risorsa. Quando aggiungi, rimuovi o sposti degli account all'interno dell'organizzazione, le policy che includono la chiave `aws:SourceOrgID` o `aws:SourceOrgPaths` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

Per quanto riguarda le politiche di attendibilità dei non-service-linked ruoli, ogni servizio incluso nella politica di fiducia ha eseguito l'`iam:PassRole`azione necessaria per verificare che il ruolo si trovi nello stesso account del servizio chiamante. Di conseguenza, l'utilizzo di `aws:SourceAccount`, `aws:SourceOrgID` o `aws:SourceOrgPaths` con tali policy di attendibilità non è necessario. L'utilizzo di `aws:SourceArn` in una politica di affidabilità consente di specificare le risorse di cui è possibile assumere un ruolo per conto, ad esempio una funzione Lambda ARN. Alcuni Servizi AWS utilizzano politiche di fiducia `aws:SourceAccount` e `aws:SourceArn` attendibilità per i ruoli appena creati, ma l'utilizzo delle chiavi non è necessario per i ruoli esistenti nel tuo account.

> **Note**
>
> Servizi AWS che si integrano con AWS Key Management Service l'utilizzo delle concessioni di chiavi KMS non supportano le chiavi `aws:SourceArn`aws:SourceAccount,`aws:SourceOrgID`, o `aws:SourceOrgPaths` condition. L'utilizzo di queste chiavi di condizione in una politica delle chiavi KMS comporterà un comportamento imprevisto se la chiave viene utilizzata anche Servizi AWS tramite concessioni di chiavi KMS.

Prevenzione sostitutiva confusa tra diversi servizi per AWS Security Token Service

Molti AWS servizi richiedono l'utilizzo di ruoli per consentire al servizio di accedere alle risorse di un altro servizio per conto dell'utente. Un ruolo che un servizio assume per eseguire operazioni a tuo nome viene chiamato ruolo del servizio. Un ruolo richiede due policy: una policy di attendibilità del ruolo, che specifica il principale a cui è consentito assumere il ruolo, e una policy delle autorizzazioni, che specifica le operazioni da eseguire con il ruolo. Una policy di attendibilità del ruolo è l'unico tipo di policy basata sulle risorse in IAM. Altri Servizi AWS hanno politiche basate sulle risorse, come una policy sui bucket di Amazon S3.

Quando un servizio assume un ruolo per tuo conto, il principale del servizio deve essere autorizzato a svolgere l'operazione `sts:AssumeRole` nella policy di attendibilità del ruolo. Quando un servizio chiama`sts:AssumeRole`, AWS STS restituisce un set di credenziali di sicurezza temporanee che il responsabile del servizio utilizza per accedere alle risorse consentite dalla politica di autorizzazione

del ruolo. Quando un servizio assume un ruolo nel tuo account, puoi includere le chiavi di contesto delle condizioni globali `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` o `aws:SourceOrgPaths` nella policy di attendibilità del ruolo per limitare l'accesso al ruolo solo alle richieste generate dalle risorse previste.

Ad esempio, in AWS Systems Manager Incident Manager, è necessario scegliere un ruolo per consentire a Incident Manager di eseguire un documento di automazione Systems Manager per conto dell'utente. Il documento di automazione può includere piani di risposta automatici per incidenti avviati da CloudWatch allarmi o eventi. EventBridge Nell'esempio seguente della policy di attendibilità del ruolo, è possibile utilizzare la chiave di condizione `aws:SourceArn` per limitare l'accesso al ruolo di servizio in base all'ARN del registro degli incidenti. Solo i registri degli incidenti creati dalla risorsa del piano di risposta `myresponseplan`sono in grado di utilizzare questo ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "ssm-incidents.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ssm-incidents:*:111122223333:incident-
record/myresponseplan/*"
      }
    }
  }
}
```

> **Note**
>
> Non tutte le integrazioni di servizi con chiavi di AWS STS supporto `aws:SourceArn` o di `aws:SourceAccount` condizione`aws:SourceOrgID`. `aws:SourceOrgPaths` L'utilizzo di queste chiavi nelle policy di attendibilità IAM con integrazioni non supportate può causare comportamenti imprevisti.

Fornire l'accesso a utenti autenticati esternamente (federazione delle identità)

I tuoi utenti potrebbero già avere identità esterne AWS, ad esempio nella tua directory aziendale. Se tali utenti devono utilizzare AWS risorse (o utilizzare applicazioni che accedono a tali risorse), devono utilizzare anche credenziali AWS di sicurezza. È possibile utilizzare un ruolo IAM per specificare le autorizzazioni per gli utenti la cui identità è federata dalla propria organizzazione o da un provider di identità di terze parti (IdP).

> **Note**
>
> Come best practice di sicurezza, ti consigliamo di gestire l'accesso degli utenti in Centro identità IAM con la federazione delle identità anziché creare utenti IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

Federazione di utenti di una applicazione per dispositivi mobili o basata sul Web con Amazon Cognito

Se crei un'app mobile o basata sul Web che accede alle AWS risorse, l'app necessita di credenziali di sicurezza per poter effettuare richieste programmatiche a. AWS Per la maggior parte degli scenari relativi alle applicazioni per dispositivi mobili, consigliamo di utilizzare Amazon Cognito. Puoi utilizzare questo servizio con AWS Mobile SDK per iOS e Mobile SDK per Android e AWS Fire OS per creare identità uniche per gli utenti e autenticarli per un accesso sicuro alle tue risorse. AWS Amazon Cognito supporta gli stessi provider di identità come quelli elencati nella sezione successiva e supporta anche identità autenticate dallo sviluppatore e accesso non autenticato (ospite). Amazon Cognito fornisce inoltre operazioni API per la sincronizzazione dei dati utente in modo che vengano conservati quando gli utenti passano da un dispositivo all'altro. Per ulteriori informazioni, consulta Utilizzo di Amazon Cognito per applicazioni per dispositivi mobili.

Federazione degli utenti con provider di servizi di identità pubblica o OpenID Connect

Quando possibile, utilizza Amazon Cognito per scenari di applicazioni per dispositivi mobili o basate sul Web. Amazon Cognito si occupa della maggior parte del behind-the-scenes lavoro con i servizi di provider di identità pubblici per te. Lavora con gli stessi servizi di terze parti e supporta anche gli accessi anonimi. Tuttavia, per ulteriori scenari avanzati, è possibile lavorare direttamente con un servizio di terze parti, ad esempio Login with Amazon, Facebook, Google o qualsiasi IdP compatibile con OpenID Connect (OIDC). Per ulteriori informazioni sull'utilizzo della federazione OIDC utilizzando uno di questi servizi, consulta. Federazione OIDC

Federazione degli utenti con SAML 2.0

Se la tua organizzazione utilizza già un pacchetto software per provider di identità che supporta SAML 2.0 (Security Assertion Markup Language 2.0), puoi creare fiducia tra la tua organizzazione come provider di identità (IdP) e AWS come fornitore di servizi. Puoi quindi utilizzare SAML per fornire ai tuoi utenti il Single Sign-On federato (SSO) o l'accesso federato alle operazioni dell' AWS Management Console API di chiamata. AWS Ad esempio, se la tua azienda utilizza Microsoft Active Directory e Active Directory Federation Services, puoi effettuare la federazione utilizzando SAML 2.0. Per ulteriori informazioni sulla federazione degli utenti con SAML 2.0, consulta Federazione SAML 2.0.

Federazione degli utenti creando un'applicazione personalizzata per la gestione di identità

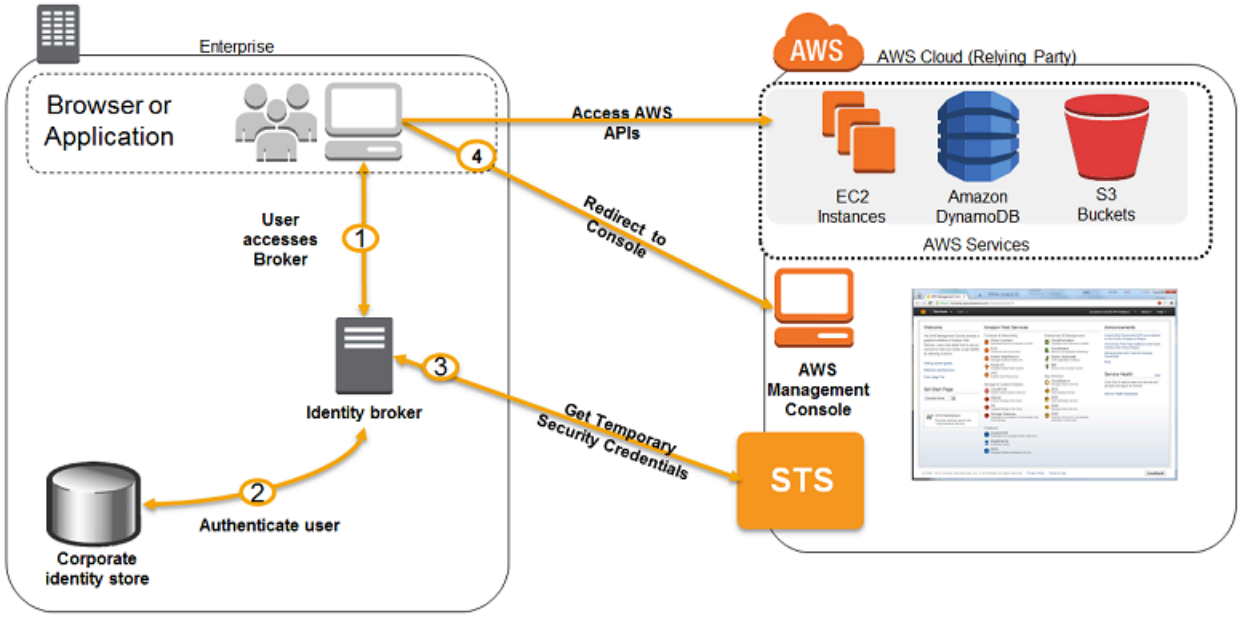
Se il proprio archivio identità non è compatibile con SAML 2.0, è possibile creare un'applicazione personalizzata per la gestione di identità per eseguire una funzione simile. L'applicazione broker autentica gli utenti, richiede credenziali temporanee per gli utenti e quindi le fornisce all'utente per accedere alle AWS risorse. AWS

Ad esempio, Example Corp. ha molti dipendenti che devono eseguire applicazioni interne che accedono alle risorse dell'azienda. AWS I dipendenti hanno già identità nel sistema di identità e autenticazione dell'azienda ed Example Corp. non desidera creare un utente IAM separato per ogni dipendente dell'azienda.

Bob è uno sviluppatore presso Example Corp. Per consentire alle applicazioni interne di Example Corp. di accedere alle AWS risorse dell'azienda, Bob sviluppa un'applicazione di identity broker personalizzata. L'applicazione verifica che i dipendenti abbiano effettuato l'accesso nel sistema di identità e autenticazione esistente, che potrebbe utilizzare LDAP, Active Directory o un altro sistema. L'applicazione del gestore identità quindi ottiene le credenziali di sicurezza provvisorie per i dipendenti. Questo scenario è simile a quello precedente (un'app mobile che utilizza un sistema di autenticazione personalizzato), tranne per il fatto che le applicazioni che richiedono l'accesso alle AWS risorse vengono eseguite tutte all'interno della rete aziendale e l'azienda dispone di un sistema di autenticazione esistente.

Per ottenere le credenziali di sicurezza provvisorie, l'applicazione del gestore identità chiama `AssumeRole` o `GetFederationToken` per ottenere le credenziali di sicurezza provvisorie, a seconda di come Bob desidera gestire le policy per gli utenti e quando scadono le credenziali provvisorie. (Per ulteriori informazioni sulle differenze tra queste operazioni API, consultare Credenziali di sicurezza temporanee in IAM e Controllo delle autorizzazioni per le credenziali di sicurezza temporanee.) La chiamata restituisce credenziali di sicurezza temporanee costituite da un

ID chiave di AWS accesso, una chiave di accesso segreta e un token di sessione. L'applicazione del gestore identità rende tali credenziali di sicurezza provvisorie disponibili all'applicazione aziendale interna. L'applicazione può quindi utilizzare le credenziali provvisorie per effettuare chiamate a AWS direttamente. L'app memorizza le credenziali finché non scadono e in seguito richiede un nuovo set di credenziali temporanee. L'immagine seguente illustra questo scenario.



Questo scenario ha i seguenti attributi:

- L'applicazione del gestore identità ha le autorizzazioni per accedere all'API di servizio token IAM (STS) per creare le credenziali di sicurezza temporanee.
- L'applicazione del gestore identità è in grado di verificare che i dipendenti siano autenticati nel sistema di autenticazione esistente.
- Gli utenti possono ottenere un URL temporaneo che consente loro di accedere alla Console di AWS gestione (denominata Single Sign-on).

Per ulteriori informazioni sulla creazione di credenziali di sicurezza provvisorie, consultare Richiesta di credenziali di sicurezza temporanee. Per ulteriori informazioni sull'accesso degli utenti federati alla console di AWS gestione, consulta. Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console

Uso di ruoli collegati ai servizi

Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a un servizio AWS . I ruoli collegati ai servizi sono predefiniti dal servizio e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per conto dell'utente. Il servizio collegato definisce anche le modalità di creazione, modifica ed eliminazione di un ruolo collegato al servizio. Un servizio può creare o eliminare automaticamente il ruolo. È possibile che ti permetta di creare, modificare o eliminare il ruolo come parte di una procedura guidata o un processo nel servizio. Oppure potrebbe richiedere l'utilizzo di IAM per creare o eliminare il ruolo. Indipendentemente dal metodo, i ruoli collegati ai servizi semplificano la procedura di configurazione di un servizio poiché non dovrai più aggiungere manualmente le autorizzazioni necessarie ai servizi per completare le operazioni per tuo conto.

> **Note**
>
> Ricorda che i ruoli di servizio sono diversi dai ruoli collegati ai servizi. Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM. Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Il servizio collegato definisce le autorizzazioni dei relativi ruoli collegati ai servizi e, a meno che non sia stato stabilito diversamente, solo quel servizio può assumere i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Prima di poter eliminare i ruoli, devi eliminare le risorse associate. Questa procedura protegge le risorse di perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

> **Tip**
>
> Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo

collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi

Per consentire a un utente o un ruolo di creare o modificare un ruolo collegato ai servizi, devi configurare le autorizzazioni per un'entità IAM (utente o ruolo).

> **Note**
>
> L'ARN per un ruolo collegato ai servizi include un'entità principale del servizio, indicata nelle policy seguenti come *SERVICE-NAME*`.amazonaws.com`. Non cercate di indovinare il principale del servizio, perché fa distinzione tra AWS maiuscole e minuscole e il formato può variare da un servizio all'altro. Per visualizzare l'entità principale di un servizio, consulta la relativa documentazione del ruolo collegato al servizio.

Per consentire a un'entità IAM di creare un ruolo specifico collegato ai servizi

Aggiungi la policy seguente a un'entità IAM che deve creare il ruolo collegato ai servizi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*",
            "Condition": {"StringLike": {"iam:AWSServiceName": "SERVICE-
NAME.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"
```

```
        }
    ]
}
```

Come consentire a un'entità IAM di creare qualunque ruolo collegato ai servizi

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve creare un ruolo collegato ai servizi o qualunque ruolo di servizio che include le policy di cui ha bisogno. Questa istruzione della policy non consente all'entità IAM di collegare una policy al ruolo.

```
{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di modificare la descrizione di qualunque ruolo di servizio

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve modificare la descrizione di un ruolo collegato ai servizi o qualunque ruolo di servizio.

```
{
    "Effect": "Allow",
    "Action": "iam:UpdateRoleDescription",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di eliminare un ruolo collegato ai servizi specifico

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve eliminare il ruolo collegato ai servizi.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-
NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"
}
```

Come consentire a un'entità IAM di eliminare qualunque ruolo collegato ai servizi

Aggiungi la seguente istruzione alla policy delle autorizzazioni per l'entità IAM che deve eliminare un ruolo collegato ai servizi ma non il ruolo di servizio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

Come consentire a un'entità IAM di passare un ruolo esistente al servizio

Alcuni AWS servizi consentono di trasferire un ruolo esistente al servizio, anziché creare un nuovo ruolo collegato al servizio. Per eseguire questa operazione, un utente deve disporre delle autorizzazioni per passare il ruolo al servizio. Aggiungi l'istruzione seguente alla policy delle autorizzazioni per l'entità IAM che deve passare un ruolo. Questa istruzione della policy consente anche all'entità di visualizzare un elenco di ruoli da cui è possibile scegliere il ruolo da passare. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS.

```
{
  "Sid": "PolicyStatementToAllowUserToListRoles",
  "Effect": "Allow",
  "Action": ["iam:ListRoles"],
  "Resource": "*"
},
{
  "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",
  "Effect": "Allow",
  "Action": [ "iam:PassRole" ],
  "Resource": "arn:aws:iam::account-id:role/my-role-for-XYZ"
}
```

Autorizzazioni indirette con ruoli collegati al servizio

Le autorizzazioni concesse da un ruolo collegato ai servizi possono essere indirettamente trasferite ad altri utenti e ruoli. Quando un ruolo collegato al servizio viene utilizzato da un AWS servizio, tale

ruolo può utilizzare le proprie autorizzazioni per chiamare altri servizi. AWS Ciò significa che gli utenti e i ruoli con le autorizzazioni per chiamare un servizio che utilizza un ruolo collegato al servizio possono avere accesso indiretto ai servizi a cui può accedere quel ruolo collegato al servizio.

Ad esempio, quando crei un'istanza database Amazon RDS, un ruolo collegato ai servizi per RDS viene creato automaticamente se non ne esiste già uno. Questo ruolo collegato al servizio consente a RDS di chiamare Amazon EC2, Amazon SNS, Amazon CloudWatch Logs e Amazon Kinesis per tuo conto. Se consenti agli utenti e ai ruoli del tuo account di modificare o creare database RDS, potrebbero interagire indirettamente con Amazon EC2, Amazon SNS, i log di Amazon Logs e le risorse CloudWatch Amazon Kinesis chiamando RDS, poiché RDS utilizzerebbe il suo ruolo collegato ai servizi per accedere a tali risorse.

Creazione di un ruolo collegato ai servizi

Il metodo utilizzato per creare un ruolo collegato ai servizi dipende dal servizio. In alcuni casi, non devi creare manualmente un ruolo collegato ai servizi. Ad esempio, quando completi un'azione specifica (ad esempio la creazione di una risorsa) nel servizio, il servizio potrebbe creare il ruolo collegato ai servizi per te. O se stavi utilizzando un servizio prima di iniziare il supporto ai ruoli collegati ai servizi, allora il servizio potrebbe aver creato automaticamente il ruolo nel tuo account. Per ulteriori informazioni, consulta Un nuovo ruolo appare nell'account AWS.

In altri casi, il servizio può supportare la creazione di un ruolo collegato ai servizi manualmente utilizzando la console di servizio, le API o la CLI. Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Per scoprire se il servizio supporta la creazione del ruolo collegato ai servizi, selezionare il link Sì per visualizzare il ruolo collegato ai servizi per quel servizio.

Se il servizio non supporta la creazione del ruolo, è possibile utilizzare IAM per creare il ruolo collegato ai servizi.

> **Important**
>
> I ruoli collegati ai servizi vengono conteggiati nel limite dei Ruoli IAM in un Account AWS, ma se è stato raggiunto il limite puoi sempre creare i ruoli collegati ai servizi nel tuo account. Solo i ruoli collegati ai servizi possono superare il limite.

Creazione di un ruolo collegato ai servizi (console)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi; inoltre, scopri se è possibile creare il ruolo dalla console del servizio, dall'API o dalla CLI.

Come creare un ruolo collegato ai servizi (console)

1. Accedi e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/. AWS Management Console
2. Nel pannello di navigazione della console IAM seleziona Ruoli. Quindi seleziona Create role (Crea ruolo).
3. Scegli il tipo di ruolo di servizio AWS .
4. Scegli il caso d'uso per il servizio. I casi d'uso sono definiti dal servizio in modo da includere la policy di attendibilità richiesta dal servizio. Quindi, seleziona Next (Successivo).
5. Scegli una o più policy di autorizzazione da collegare al ruolo. A seconda del caso d'uso selezionato, il servizio può eseguire una di queste operazioni:

   - Definire le autorizzazioni utilizzate dal ruolo.
   - Consentire di scegliere tra un set limitato di autorizzazioni.
   - Consentire di scegliere qualsiasi autorizzazione.
   - Ti consente di non selezionare policy in questo momento, creare le policy successivamente e quindi collegarle al ruolo.

   Seleziona la casella di controllo accanto alla policy che assegna le autorizzazioni desiderate per il ruolo, quindi scegli Next (Successivo).

   > **Note**
   >
   > Le autorizzazioni specificate sono disponibili per qualsiasi entità che utilizza il ruolo. Per default, un ruolo non dispone di autorizzazioni.

6. Il grado di personalizzazione per Nome ruolo viene definito dal servizio. Se il servizio definisce il nome del ruolo, allora questa opzione non può essere modificata. In altri casi, il servizio può definire un prefisso per il ruolo e consentirti di inserire un suffisso opzionale.

Se possibile, inserisci il suffisso del nome del ruolo da aggiungere al nome predefinito. Il suffisso consente di identificare lo scopo del ruolo. I nomi dei ruoli devono essere univoci all'interno dell'account AWS . Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **<service-linked-role-name>_SAMPLE** che **<service-linked-role-name>_sample**. Poiché varie entità possono fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo averlo creato.

7. (Facoltativo) In Description (Descrizione), modifica la descrizione per il nuovo ruolo collegato ai servizi.
8. Non è possibile collegare tag ai ruoli collegati ai servizi durante la creazione. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
9. Rivedere il ruolo e scegliere Crea ruolo.

Creazione di un ruolo collegato ai servizi (AWS CLI)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi e se è possibile creare il ruolo dalla CLI del servizio. Se la CLI del servizio non è supportata, puoi usare i comandi IAM per creare un ruolo collegato ai servizi con la policy di attendibilità e le policy in linea che il servizio richiede per assumere il ruolo.

Per creare un ruolo collegato ai servizi (AWS CLI)

Esegui il comando seguente:

```
aws iam create-service-linked-role --aws-service-name SERVICE-NAME.amazonaws.com
```

Creazione di un ruolo collegato ai servizi (API AWS)

Prima di creare un ruolo collegato ai servizi in IAM, scopri se il servizio collegato crea automaticamente i ruoli collegati ai servizi e scopri se è possibile creare il ruolo dalle API del servizio. Se l'API del servizio non è supportata, puoi utilizzarla AWS per creare un ruolo collegato al servizio con la policy di fiducia e le politiche in linea necessarie al servizio per assumere il ruolo.

Per creare un ruolo collegato al servizio (API)AWS

Utilizzare la chiamata API CreateServiceLinkedRole. Nella richiesta, specificare un nome del servizio di *SERVICE_NAME_URL*.amazonaws.com.

Ad esempio, per creare il ruolo collegato ai servizi Lex Bots (Bot di Lex), utilizzare `lex.amazonaws.com`.

Modificare un ruolo collegato ai servizi

Il metodo utilizzato per modificare un ruolo collegato ai servizi dipende dal servizio. Alcuni servizi consentono di modificare le autorizzazioni per un ruolo collegato ai servizi dalla console di servizio, dalle API o dalla CLI. Tuttavia, dopo aver creato un ruolo collegato ai servizi, non è possibile modificare il nome del ruolo poiché varie entità possono farvi riferimento. Puoi modificare la descrizione di qualsiasi ruolo dalla console IAM, dall'API o dalla CLI.

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Per scoprire se il servizio supporta la modifica del ruolo collegato ai servizi, selezionare il link Sì per visualizzare il ruolo collegato ai servizi per quel servizio.

Modifica della descrizione di un ruolo collegato ai servizi (console)

Puoi utilizzare la console IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
2. Scegliere il nome del ruolo da modificare.
3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).
4. Digita una nuova descrizione nella casella e scegli Save (Salva).

Modifica della descrizione di un ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS CLI per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza i seguenti comandi:

```
aws iam get-role --role-name ROLE-NAME
```

Per fare riferimento ai ruoli con i comandi della CLI utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.

2. Per aggiornare la descrizione di un ruolo collegato ai servizi, utilizza il seguente comando:

```
aws iam update-role --role-name ROLE-NAME --description OPTIONAL-DESCRIPTION
```

Modifica della descrizione di un ruolo collegato al servizio (API)AWS

È possibile utilizzare l' AWS API per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato al servizio (API)AWS

1. (Facoltativo) Per visualizzare l'attuale descrizione per un ruolo, effettua una chiamata all'operazione seguente e specifica il nome del ruolo:

   AWS API: GetRole
2. Per aggiornare la descrizione di un ruolo, effettua una chiamata all'operazione seguente e specifica il nome (e facoltativamente la descrizione) del ruolo:

   AWS API: UpdateRole

Eliminazione del ruolo collegato ai servizi

Il metodo utilizzato per creare un ruolo collegato ai servizi dipende dal servizio. In alcuni casi, non devi eliminare manualmente un ruolo collegato ai servizi. Ad esempio, quando completi un'operazione specifica (come eliminare una risorsa) nel servizio, il servizio potrebbe eliminare il ruolo collegato ai servizi per te.

In altri casi, il servizio può supportare l'eliminazione di un ruolo collegato ai servizi manualmente dalla console del servizio, dall'API o dalla AWS CLI.

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Per scoprire se il servizio supporta l'eliminazione del ruolo collegato ai servizi, scegli il link Sì per visualizzare il ruolo collegato ai servizi per quel servizio.

Se il servizio non supporta l'eliminazione del ruolo, puoi eliminare il ruolo collegato al servizio dalla console IAM, dall'API o. AWS CLI Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare quel ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto verificare che il ruolo non abbia sessioni attive ed eliminare tutte le risorse utilizzate dal ruolo.

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Ruoli. Quindi, scegli il nome (non la casella di controllo) del ruolo collegato al servizio.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, scegliere la scheda Access Advisor (Consulente accessi).
4. Nella scheda Access Advisor (Consulente accessi), esamina l'attività recente per il ruolo collegato ai servizi.

> **Note**
>
> Se non sei certo che il servizio stia utilizzando il ruolo collegato ai servizi, puoi provare a eliminare il ruolo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni in cui il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato al servizio.

Per rimuovere le risorse utilizzate dal ruolo collegato ai servizi

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta la pagina AWS servizi che funzionano con IAM e cerca i servizi per cui è indicato Sì nella colonna Ruolo collegato ai servizi. Per scoprire se il servizio supporta l'eliminazione del ruolo collegato ai servizi, scegli il link Sì per

visualizzare il ruolo collegato ai servizi per quel servizio. Consulta la documentazione per quel servizio per Scopri come rimuovere le risorse utilizzate dal ruolo collegato ai servizi.

Eliminazione di un ruolo collegato ai servizi (console)

Puoi utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli). Quindi, seleziona la casella di controllo accanto al nome del ruolo che desideri eliminare, non il nome o la riga stessa.
3. In Operazioni ruolo nella parte superiore della pagina, seleziona Elimina.
4. Nella finestra di dialogo di conferma, esamina le informazioni relative all'ultimo accesso, che mostrano l'ultima volta che ciascuno dei ruoli selezionati ha effettuato l'ultimo accesso a un AWS servizio. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, seleziona Yes, Delete (Sì, elimina) per richiedere l'eliminazione del ruolo collegato ai servizi.
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno.

   - Se il task viene eseguito correttamente, il ruolo viene rimosso dall'elenco e nella parte superiore della pagina viene visualizzata una notifica di completamento.
   - Se il task non viene eseguito correttamente, puoi scegliere View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

     > **Note**
     >
     > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione

non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna.

- Se il task non viene eseguito e la notifica non include un elenco di risorse, il servizio potrebbe non restituire questa informazione. Per scoprire come eliminare le risorse per quel servizio, consultare la pagina AWS servizi che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione del ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS CLI per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (AWS CLI)

1. Se conosci il nome del ruolo collegato ai servizi da eliminare, inserisci il comando seguente per elencare i ruoli nell'account:

   ```
   aws iam get-role --role-name role-name
   ```

   Per fare riferimento ai ruoli con i comandi della CLI utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.
2. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id`dalla risposta per controllare lo stato del task di eliminazione. Inserisci il seguente comando per inviare una richiesta di eliminazione di un ruolo collegato ai servizi:

   ```
   aws iam delete-service-linked-role --role-name role-name
   ```
3. Inserisci il seguente comando per verificare lo stato dell'attività di eliminazione:

   ```
   aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita

perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

> **Note**
>
> In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna. Per scoprire come eliminare le risorse per un servizio che non restituisce nessuna risorsa, consultare la pagina AWS servizi che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione di un ruolo collegato ai servizi (API AWS)

Puoi utilizzare l' AWS API per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato al servizio (API)AWS

1. Per inviare una richiesta di eliminazione per un ruolo collegato al servizio, chiama. DeleteServiceLinkedRole Nella richiesta, specifica il nome del ruolo.

   Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId`dalla risposta per controllare lo stato del task di eliminazione.
2. Per verificare lo stato dell'eliminazione, chiama. GetServiceLinkedRoleDeletionStatus Nella richiesta, specificare il `DeletionTaskId`.

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se

il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

> **Note**
>
> In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna. Per scoprire come eliminare le risorse per un servizio che non restituisce nessuna risorsa, consultare la pagina AWS servizi che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Creazione di ruoli IAM

Per creare un ruolo, puoi utilizzare l' AWS Management Console AWS CLI API Tools for Windows PowerShell o IAM.

Se utilizzi il AWS Management Console, una procedura guidata ti guida attraverso i passaggi per la creazione di un ruolo. La procedura guidata prevede passaggi leggermente diversi a seconda che si stia creando un ruolo per un AWS servizio, per un utente o per un Account AWS utente federato.

Argomenti

- Creazione di un ruolo per delegare le autorizzazioni a un utente IAM
- Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS
- Creazione di un ruolo per un provider di identità di terza parte (federazione)
- Creazione di un ruolo utilizzando policy di attendibilità personalizzate (console)
- Esempi di policy per la delega dell'accesso

Creazione di un ruolo per delegare le autorizzazioni a un utente IAM

Puoi utilizzare i ruoli IAM per delegare l'accesso alle tue AWS risorse. Con i ruoli IAM, puoi stabilire relazioni di fiducia tra il tuo account fiduciario e altri account AWS affidabili. L'account che concede

fiducia possiede la risorsa alla quale accedere e l'account affidabile contiene gli utenti che devono accedere alla risorsa. Tuttavia, è possibile che un altro account sia proprietario di una risorsa nell'account in uso. L'account che concede fiducia potrebbe infatti consentire all'account attendibile di creare nuove risorse, ad esempio creando nuovi oggetti in un bucket Amazon S3. In tal caso, l'account che crea la risorsa ne è proprietario e controlla chi può accedervi.

Dopo aver creato la relazione di fiducia, un utente IAM o un'applicazione dell'account affidabile può utilizzare l'operazione `AssumeRole`API AWS Security Token Service (AWS STS). Questa operazione fornisce credenziali di sicurezza temporanee che consentono l'accesso alle AWS risorse del tuo account.

Gli account possono essere controllati da sé stessi oppure l'account con gli utenti può essere controllato da terze parti. Se l'altro account con gli utenti è un account Account AWS che non controlli, puoi utilizzare l'`externalId`attributo. L'ID esterno può essere qualsiasi parola o numero concordato tra l'utente e l'amministratore dell'account di terze parti. Questa opzione aggiunge automaticamente una condizione alla policy di affidabilità che consente all'utente di assumere il ruolo solo se la richiesta include il corretto `sts:ExternalID`. Per ulteriori informazioni, consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.

Per informazioni su come utilizzare i ruoli per delegare le autorizzazioni, consultare Termini e concetti dei ruoli. Per informazioni sull'utilizzo di un ruolo di servizio per consentire l'accesso a risorse nel proprio account, consultare Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS.

Creazione di un ruolo IAM (console)

Puoi utilizzare il AWS Management Console per creare un ruolo che un utente IAM può assumere. Ad esempio, supponiamo che l'organizzazione disponga di più Account AWS elementi per isolare un ambiente di sviluppo da un ambiente di produzione. Per informazioni di alto livello sulla creazione di un ruolo che consenta agli utenti nell'account di sviluppo di accedere alle risorse nell'account di produzione, consulta la sezione Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati.

Per creare un ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, selezionare Roles (Ruoli) e Crea ruolo.
3. Scegli il tipo di ruolo Account AWS.

4. Per creare un ruolo per il tuo account, scegli This account (Questo account). Per creare un ruolo per un altro account, scegli Altro Account AWS e inserisci l'ID account ID al quale desideri concedere l'accesso alle risorse.

   L'amministratore dell'account specificato può concedere l'autorizzazione di assumere questo ruolo a qualsiasi utente IAM in tale account. Per eseguire questa operazione, l'amministratore collega una policy all'utente o al gruppo che garantisce l'autorizzazione per l'operazione `sts:AssumeRole`. Tale policy deve specificare il nome ARN del ruolo `Resource`.
5. Per concedere le autorizzazioni agli utenti da un account di cui non hai il controllo e se tali utenti assumeranno il ruolo a livello di programmazione, seleziona Require external ID (Richiedi ID esterno). L'ID esterno può essere qualsiasi parola o numero concordato tra l'utente e l'amministratore dell'account di terze parti. Questa opzione aggiunge automaticamente una condizione alla policy di affidabilità che consente all'utente di assumere il ruolo solo se la richiesta include il corretto `sts:ExternalID`. Per ulteriori informazioni, consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.

   > **Important**
   >
   > La scelta di questa opzione limita l'accesso al ruolo solo tramite Tools for Windows PowerShell o l' AWS API. AWS CLI Questo perché non è possibile utilizzare la AWS console per passare a un ruolo che presenta una `externalId` condizione nella politica di attendibilità. Tuttavia, è possibile creare questo tipo di accesso a livello di codice scrivendo uno script o un'applicazione utilizzando il kit SDK rilevante. Per ulteriori informazioni e uno script di esempio, vedi Come abilitare l'accesso da più account a AWS Management Console nel blog sulla AWS sicurezza.
6. Se si desidera limitare il ruolo agli utenti che accedono con la multi-factor authentication (MFA), selezionare Require MFA (Richiedi MFA). Questa opzione aggiunge una condizione alla policy di affidabilità del ruolo che controlla un accesso MFA. Gli utenti che desidera assumere il ruolo deve effettuare l'accesso temporaneo con una password una tantum temporanea configurata da un dispositivo MFA. Gli utenti senza l'autenticazione MFA non possono assumere il ruolo. Per ulteriori informazioni sulla funzionalità MFA, consultare Utilizzo dell'autenticazione a più fattori (MFA) in AWS.
7. Seleziona Successivo.
8. IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Selezionare la policy delle autorizzazioni da utilizzare o scegliere Crea policy per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta

Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Selezionare la casella di controllo accanto alle policy di autorizzazione da assegnare a chiunque assuma il ruolo. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.

9. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

   Aprire la sezione Set permissions boundary (Imposta limite delle autorizzazioni) e selezionare Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.

10. Seleziona Successivo.

11. In Nome ruolo, immetti un nome per il ruolo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole. Quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante la procedura di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole. Poiché varie entità possono fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.

12. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.

13. Scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni) per modificare i casi d'uso e le autorizzazioni per il ruolo. Verrai reindirizzato alle pagine precedenti per apportare le modifiche.

14. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.

15. Rivedere il ruolo e scegliere Crea ruolo.

> **Important**
>
> Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire ai singoli utenti nell'account attendibile l'autorizzazione a passare al ruolo nella console o ad assumere il ruolo a livello di codice. Per ulteriori informazioni su questa fase, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

Creazione di un ruolo IAM (AWS CLI)

La creazione di un ruolo da AWS CLI richiede più passaggi. Quando si utilizza la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con la console è AWS CLI necessario eseguire ogni passaggio in modo esplicito e autonomo. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo per accesso tra account (AWS CLI)

1. Creare un ruolo: aws iam create-role
2. Allega una politica di autorizzazioni gestite al ruolo: aws iam attach-role-policy

   oppure

   Crea una politica di autorizzazioni in linea per il ruolo: aws iam put-role-policy
3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM.
4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

L'esempio seguente mostra i primi due passaggi, più comuni, per la creazione di un ruolo per più account in un ambiente semplice. Questo esempio permette agli utenti dell'account `123456789012` di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. L'esempio presuppone inoltre l'uso un computer client con Windows e che l'interfaccia a riga di comando sia già configurata con le credenziali dell'account e la regione. Per ulteriori informazioni, vedere Configurazione dell'interfaccia a AWS riga di comando.

In questo esempio, è necessario includere la seguente policy di attendibilità nel primo comando al momento della creazione del ruolo. Questa policy di attendibilità consente agli utenti dell'account `123456789012` di assumere il ruolo tramite l'operazione `AssumeRole`, ma solo se l'utente fornisce l'autenticazione MFA utilizzando i parametri `SerialNumber` e `TokenCode`. Per ulteriori informazioni sulla funzionalità MFA, consultare Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
          "Action": "sts:AssumeRole",
          "Condition": { "Bool": { "aws:MultiFactorAuthPresent": "true" } }
      }
  ]
}
```

Important

Se l'elemento `Principal` contiene l'ARN per un determinato utente o ruolo IAM, quando la policy viene salvata l'ARN viene trasformato in un ID principale univoco. Ciò aiuta a mitigare il rischio che qualcuno aumenti le proprie autorizzazioni rimuovendo e ricreando il ruolo o l'utente. Questo ID non è normalmente presente nella console, perché avviene anche una trasformazione inversa nell'ARN quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina il ruolo o l'utente, l'ID principale viene visualizzato nella console perché non è più AWS possibile mapparlo su un ARN. Pertanto, se si elimina e crea nuovamente un utente o un ruolo a cui viene fatto riferimento in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire l'ARN.

Quando si utilizza il secondo comando, è necessario collegare una policy gestita esistente al ruolo. La policy delle autorizzazioni seguente consente agli utenti che assumono il ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": "s3:ListBucket",
          "Resource": "arn:aws:s3:::example_bucket"
      }
  ]
}
```

Per creare questo ruolo `Test-UserAccess-Role`, è prima necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforacct123456789012.json` nella cartella `policies` dell'unità `C:` locale. Quindi salva la politica di autorizzazione precedente come politica gestita dai clienti nel tuo account Account AWS con il nome. `PolicyForRole` È quindi possibile utilizzare i comandi seguenti per creare il ruolo e collegare la policy gestita.

```
# Create the role and attach the trust policy file that allows users in the specified
 account to assume the role.
$ aws iam create-role --role-name Test-UserAccess-Role --assume-role-policy-document
 file://C:\policies\trustpolicyforacct123456789012.json

# Attach the permissions policy (in this example a managed policy) to the role to
 specify what it is allowed to do.
$ aws iam attach-role-policy --role-name Test-UserAccess-Role --policy-arn
 arn:aws:iam::123456789012:policy/PolicyForRole
```

> **Important**
>
> Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire a singoli utenti nell'account affidabile le autorizzazioni per passare al ruolo. Per ulteriori informazioni su questa fase, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

Dopo aver creato il ruolo e avergli concesso le autorizzazioni per eseguire AWS attività o accedere alle AWS risorse, qualsiasi utente dell'123456789012account può assumere il ruolo. Per ulteriori informazioni, consulta Passaggio a un ruolo IAM (AWS CLI).

Creazione di un ruolo IAM (AWS API)

La creazione di un ruolo dall' AWS API prevede diversi passaggi. Quando si usa la console per creare un ruolo, molti dei passaggi vengono eseguiti automaticamente, ma con l'API ogni passaggio deve essere eseguito esplicitamente dall'utente. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Creare un ruolo nel codice (AWS API)

1. Crea un ruolo: CreateRole

   Per la policy di affidabilità del ruolo, è possibile specificare una posizione del file.

2. Allega una politica di autorizzazione gestita al ruolo: AttachRolePolicy

   oppure

   Crea una politica di autorizzazione in linea per il ruolo: PutRolePolicy

   > **Important**
   >
   > Ricordare che questa è solo la prima metà della configurazione obbligatoria. È inoltre necessario fornire a singoli utenti nell'account affidabile le autorizzazioni per passare al ruolo. Per ulteriori informazioni su questa fase, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

3. (Facoltativo) Aggiungi attributi personalizzati all'utente allegando tag: TagRole

   Per ulteriori informazioni, consulta Gestione dei tag sugli utenti IAM (AWS CLI o AWS API).

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Dopo aver creato il ruolo e avergli concesso le autorizzazioni per eseguire AWS attività o accedere alle AWS risorse, è necessario concedere le autorizzazioni agli utenti dell'account per consentire loro di assumere il ruolo. Per ulteriori informazioni sull'assunzione di un ruolo, consulta Passaggio a un ruolo IAM (AWS API).

Creazione di un ruolo IAM (AWS CloudFormation)

Per informazioni sulla creazione di un ruolo IAM in AWS CloudFormation, consulta il riferimento alle risorse e alle proprietà e gli esempi nella Guida per l'AWS CloudFormation utente.

Per ulteriori informazioni sui modelli IAM in AWS CloudFormation, consulta gli snippet di AWS Identity and Access Management modello nella Guida per l'AWS CloudFormation utente.

Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS

Molti AWS servizi richiedono l'utilizzo di ruoli per consentire al servizio di accedere alle risorse di altri servizi per conto dell'utente. Un ruolo che un servizio assume per eseguire operazioni a tuo nome viene chiamato ruolo del servizio. Quando un ruolo fornisce un determinato scopo per un servizio, viene categorizzato come ruolo del servizio per le istanze EC2 (per esempio) oppure come ruolo

collegato al servizio. Per visualizzare i servizi che supportano ruoli collegati ai servizi, oppure se un servizio supporta qualsiasi forma di credenziali provvisorie, consulta AWS servizi che funzionano con IAM. Per apprendere come un singolo servizio utilizza i ruoli, scegli il nome del servizio nella tabella e visualizza la documentazione relativa a tale servizio.

Quando imposti l'`PassRole`autorizzazione, devi assicurarti che a un utente non venga assegnato un ruolo in cui il ruolo dispone di più autorizzazioni di quelle che desideri che l'utente abbia. Ad esempio, Alice potrebbe non essere autorizzata a eseguire alcuna azione su Amazon S3. Se Alice potesse trasferire un ruolo a un servizio che consente le azioni di Amazon S3, il servizio potrebbe eseguire azioni Amazon S3 per conto di Alice durante l'esecuzione del job.

Per informazioni su come i ruoli aiutano a delegare le autorizzazioni, consulta Termini e concetti dei ruoli.

Autorizzazioni del ruolo del servizio

Per consentire a una entità IAM (utente o ruolo) di creare o modificare un ruolo di servizio, occorre configurare le autorizzazioni.

> **Note**
>
> L'ARN per un ruolo collegato ai servizi include un principale del servizio, indicata nelle policy seguenti come *`SERVICE-NAME`*`.amazonaws.com`. Non tentare di indovinare il principale del servizio, perché fa distinzione tra maiuscole e minuscole e il formato può variare tra i servizi AWS . Per visualizzare l'entità principale di un servizio, consulta la relativa documentazione del ruolo collegato al servizio.

Come consentire a un'entità IAM di creare un ruolo di servizio specifico

Aggiungi la policy seguente all'entità IAM che deve creare il ruolo di servizio. Questa policy ti permette di creare un ruolo del servizio per il servizio specificato e utilizzando un nome specifico. Puoi quindi collegare le policy gestite o inline a tale ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "iam:AttachRolePolicy",
                "iam:CreateRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
        }
    ]
}
```

Come consentire a un'entità IAM di creare un qualsiasi ruolo di servizio

AWS consiglia di consentire solo agli utenti amministrativi di creare qualsiasi ruolo di servizio. Una persona con autorizzazioni per creare un ruolo e allegare qualsiasi policy può eseguire l'escalation delle proprie autorizzazioni. Invece, crea una policy che consenta a questa persona di creare solo i ruoli di cui hanno bisogno o lascia che un amministratore crei il ruolo di servizio per suo conto.

Per allegare una policy che consenta a un amministratore di accedere all'intero account Account AWS, utilizza la policy AdministratorAccess AWS gestita.

Come consentire a un'entità IAM di modificare un ruolo di servizio

Aggiungi la policy seguente all'entità IAM che deve modificare il ruolo di servizio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EditSpecificServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:PutRolePolicy",
                "iam:UpdateRole",
                "iam:UpdateRoleDescription"
            ],
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
        },
```

```
        {
            "Sid": "ViewRolesAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicy",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

Come consentire a un'entità IAM di eliminare un ruolo di servizio specifico

Aggiungi l'istruzione seguente alla policy delle autorizzazioni per l'entità IAM che deve eliminare il ruolo di servizio specificato.

```
{
    "Effect": "Allow",
    "Action": "iam:DeleteRole",
    "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
}
```

Come consentire a un'entità IAM di eliminare qualunque ruolo di servizio

AWS consiglia di consentire solo agli utenti amministrativi di eliminare qualsiasi ruolo di servizio. Invece, crea una policy che consenta loro di eliminare solo i ruoli di cui hanno bisogno o lascia che un amministratore elimini il ruolo di servizio per suo conto.

Per allegare una politica che consenta a un amministratore di accedere all'intero account Account AWS, utilizza la politica AdministratorAccess AWS gestita.

Creazione di un ruolo per un AWS servizio (console)

È possibile utilizzare il AWS Management Console per creare un ruolo per un servizio. Dal momento che alcuni servizi supportano più ruoli del servizio, consulta la documentazione AWS relativa al servizio per determinare quale caso d'uso selezionare. È possibile apprendere come assegnare le necessarie policy di affidabilità e autorizzazioni al ruolo, in modo che il servizio possa assumere quel ruolo per conto dell'utente. Le operazioni che è possibile utilizzare per controllare le autorizzazioni per il tuo ruolo possono variare, a seconda del modo in cui il servizio definisce i casi d'uso e della creazione o meno di un ruolo collegato ai servizi.

Per creare un ruolo per una Servizio AWS (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Servizio o caso d'uso, scegli un servizio, quindi scegli il caso d'uso. I casi d'uso sono definiti dal servizio in modo da includere la policy di attendibilità richiesta dal servizio.
5. Seleziona Successivo.
6. Per i criteri di autorizzazione, le opzioni dipendono dal caso d'uso selezionato:
   - Se il servizio definisce le autorizzazioni per il ruolo, non è possibile selezionare le politiche di autorizzazione.
   - Seleziona da un set limitato di politiche di autorizzazione.
   - Seleziona tra tutte le politiche di autorizzazione.
   - Seleziona nessuna politica di autorizzazione, crea le politiche dopo la creazione del ruolo e quindi allega le politiche al ruolo.
7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.
   a. Apri la sezione Imposta i limiti delle autorizzazioni, quindi scegli Usa un limite di autorizzazioni per controllare il numero massimo di autorizzazioni per il ruolo.

      IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.
   b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
8. Seleziona Successivo.
9. Per Role name, le opzioni dipendono dal servizio:
   - Se il servizio definisce il nome del ruolo, non è possibile modificare il nome del ruolo.
   - Se il servizio definisce un prefisso per il nome del ruolo, è possibile inserire un suffisso opzionale.
   - Se il servizio non definisce il nome del ruolo, puoi assegnare un nome al ruolo.

   > **Important**
   >
   > Quando assegnate un nome a un ruolo, tenete presente quanto segue:

- I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS account e non possono essere resi unici per caso.

  Ad esempio, non creare ruoli denominati entrambi **`PRODROLE`** e**`prodrole`**. Quando un nome di ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato dai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
- Non è possibile modificare il nome del ruolo dopo averlo creato perché altre entità potrebbero fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.
11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, nelle sezioni Passo 1: Seleziona entità attendibili o Passaggio 2: Aggiungi autorizzazioni, scegli Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consulta la sezione Applicazione di tag alle risorse IAM nella Guida per l'utente di IAM.
13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Creazione di un ruolo per un servizio (AWS CLI)

La creazione di un ruolo da AWS CLI richiede più passaggi. Quando si utilizza la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con la console AWS CLI è necessario eseguire esplicitamente ogni passaggio da soli. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Se il servizio in uso è Amazon EC2, è necessario creare anche un profilo dell'istanza e aggiungervi il ruolo. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo per un AWS servizio da AWS CLI

1. Il seguente comando `create-role` crea un ruolo denominato Ruolo di test e gli collega una policy di attendibilità:

   ```
   aws iam create-role --role-name Test-Role --assume-role-policy-document
   file://Test-Role-Trust-Policy.json
   ```
2. Allega una politica di autorizzazioni gestite al ruolo: aws iam attach-role-policy.

Ad esempio, il seguente comando `attach-role-policy` allega la policy gestita AWS denominata `ReadOnlyAccess` al ruolo IAM denominato `ReadOnlyRole`:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
ReadOnlyAccess --role-name ReadOnlyRole
```

oppure

Crea una politica di autorizzazioni in linea per il ruolo: aws iam put-role-policy

Per aggiungere una policy di autorizzazioni in linea, consulta l'esempio seguente:

```
aws iam put-role-policy --role-name Test-Role --policy-name
ExamplePolicy --policy-document file://AdminPolicy.json
```

3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM.

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Se intendi utilizzare il ruolo con Amazon EC2 o un altro AWS servizio che utilizza Amazon EC2, devi archiviare il ruolo in un profilo di istanza. Un profilo dell'istanza è un container per un ruolo che può essere associato a un'istanza Amazon EC2 quando viene avviato. Un profilo dell'istanza può contenere un solo ruolo e tale limite non può essere aumentato. Se crei il ruolo utilizzando AWS Management Console, il profilo dell'istanza viene creato per te con lo stesso nome del ruolo. Per ulteriori informazioni sui profili delle istanze, consulta Utilizzo dei profili delle istanze. Per informazioni su come avviare un'istanza EC2 con un ruolo, consulta Controlling Access to Amazon EC2 Resources nella Amazon EC2 User Guide.

Per creare un profilo dell'istanza e memorizzarvi il ruolo (AWS CLI)

1. Crea un profilo di istanza: aws iam create-instance-profile
2. Aggiungi il ruolo al profilo dell'istanza: aws iam add-role-to-instance -profile

Il comando di AWS CLI esempio riportato di seguito illustra i primi due passaggi per la creazione di un ruolo e l'assegnazione delle autorizzazioni. Mostra inoltre i due passaggi necessari per creare un profilo dell'istanza e aggiungere il ruolo al profilo. Questa policy di attendibilità di esempio permette al servizio Amazon EC2 di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. L'esempio presuppone inoltre l'uso un computer client con Windows e che l'interfaccia a riga di comando sia già configurata con le credenziali dell'account e la regione. Per ulteriori informazioni, vedere Configurazione dell'interfaccia a riga di comando. AWS

In questo esempio, è necessario includere la seguente policy di attendibilità nel primo comando al momento della creazione del ruolo. La policy di attendibilità consente al servizio Amazon EC2 di assumere il ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "ec2.amazonaws.com"},
    "Action": "sts:AssumeRole"
  }
}
```

Quando si utilizza il secondo comando, è necessario collegare una policy di autorizzazione al ruolo. L'esempio di policy di autorizzazione seguente consente al ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

Per creare questo ruolo `Test-Role-for-EC2`, è innanzitutto necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforec2.json` e la policy di autorizzazione precedente con il nome `permissionspolicyforec2.json` nella directory `policies` dell'unità `C:` locale. È quindi possibile utilizzare i comandi seguenti per creare il ruolo, collegare la policy, creare il profilo dell'istanza e aggiungere il ruolo al profilo dell'istanza.

```
# Create the role and attach the trust policy that allows EC2 to assume this role.
$ aws iam create-role --role-name Test-Role-for-EC2 --assume-role-policy-document
 file://C:\policies\trustpolicyforec2.json

# Embed the permissions policy (in this example an inline policy) to the role to
 specify what it is allowed to do.
$ aws iam put-role-policy --role-name Test-Role-for-EC2 --policy-name Permissions-
Policy-For-Ec2 --policy-document file://C:\policies\permissionspolicyforec2.json

# Create the instance profile required by EC2 to contain the role
$ aws iam create-instance-profile --instance-profile-name EC2-ListBucket-S3

# Finally, add the role to the instance profile
$ aws iam add-role-to-instance-profile --instance-profile-name EC2-ListBucket-S3 --
role-name Test-Role-for-EC2
```

Quando avvii l'istanza EC2, specifica il nome del profilo dell'istanza nella pagina Configura i dettagli dell'istanza se utilizzi la AWS console. Se utilizzi il comando della CLI `aws ec2 run-instances`, specifica il parametro `--iam-instance-profile`.

Creazione di un ruolo per un servizio (API AWS)

La creazione di un ruolo dall' AWS API prevede diversi passaggi. Quando si usa la console per creare un ruolo, molti dei passaggi vengono eseguiti automaticamente, ma con l'API ogni passaggio deve essere eseguito esplicitamente dall'utente. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Se il servizio in uso è Amazon EC2, è necessario creare anche un profilo dell'istanza e aggiungervi il ruolo. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Creare un ruolo per un AWS servizio (AWS API)

1. Crea un ruolo: CreateRole

   Per la policy di affidabilità del ruolo, è possibile specificare una posizione del file.
2. Allega una politica di autorizzazioni gestite al ruolo: AttachRole Politica

   oppure

   Crea una politica di autorizzazioni in linea per il ruolo: Policy PutRole
3. (Facoltativo) Aggiungi attributi personalizzati all'utente allegando tag: TagRole

Per ulteriori informazioni, consulta Gestione dei tag sugli utenti IAM (AWS CLI o AWS API).

4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Se intendi utilizzare il ruolo con Amazon EC2 o un altro AWS servizio che utilizza Amazon EC2, devi archiviare il ruolo in un profilo di istanza. Un profilo dell'istanza è un container per un ruolo. Ogni profilo dell'istanza può contenere un solo ruolo e tale limite non può essere superato. Se crei il ruolo in AWS Management Console, il profilo dell'istanza viene creato per te con lo stesso nome del ruolo. Per ulteriori informazioni sui profili delle istanze, consulta Utilizzo dei profili delle istanze. Per informazioni su come avviare un'istanza Amazon EC2 con un ruolo, consulta Controlling Access to Amazon EC2 Resources nella Amazon EC2 User Guide.

Per creare un profilo di istanza e memorizzare il ruolo al suo interno (API)AWS

1. Crea un profilo di istanza: CreateInstanceProfilo
2. Aggiungi il ruolo al profilo dell'istanza: AddRoleToInstanceProfilo

Creazione di un ruolo per un provider di identità di terza parte (federazione)

Puoi utilizzare provider di identità invece di creare utenti IAM nel tuo Account AWS. Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori informazioni sulla federazione e sui provider di identità, consultare Provider di identità e federazione.

Creazione di un ruolo per gli utenti federati (console)

Le procedure per la creazione di un ruolo per gli utenti federati dipendono dai provider di terze parti disponibili:

- Per OpenID Connect (OIDC), vedere. Creare un ruolo per la federazione OpenID Connect (console)
- Per SAML 2.0, consulta Creare un ruolo per la federazione SAML 2.0 (console).

Creazione di un ruolo per l'accesso federato (AWS CLI)

Le procedure per creare un ruolo per il provider di identità supportato (OIDC o SAML) dalla AWS CLI sono identiche. La differenza consiste nel contenuto della policy di affidabilità creata in passaggi preliminari. Inizia seguendo le fasi descritte nella sezione dei prerequisiti per il tipo di provider in uso:

- Per un provider OIDC, consulta Prerequisiti per la creazione di un ruolo per OIDC.
- Per un provider SAML, consulta Prerequisiti per la creazione di un ruolo per SAML.

La creazione di un ruolo da AWS CLI richiede più passaggi. Quando si utilizza la console per creare un ruolo, molti passaggi vengono eseguiti automaticamente, ma con la console AWS CLI è necessario eseguire esplicitamente ogni passaggio da soli. È necessario creare il ruolo e quindi assegnargli una policy di autorizzazione. Puoi anche scegliere di impostare il limite delle autorizzazioni per il ruolo.

Per creare un ruolo per la federazione delle identità (AWS CLI)

1. Creare un ruolo: aws iam create-role
2. Allega una politica di autorizzazioni al ruolo: aws iam attach-role-policy

   oppure

   Crea una politica di autorizzazioni in linea per il ruolo: aws iam put-role-policy
3. (Facoltativo) Aggiungere attributi personalizzati al ruolo collegando tag: aws iam tag-role

   Per ulteriori informazioni, consulta Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM.
4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: aws iam put-role-permissions-boundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

L'esempio seguente mostra i primi due passaggi, più comuni, per la creazione di un ruolo del provider di identità in un ambiente semplice. Questo esempio permette agli utenti dell'account `123456789012` di assumere il ruolo e visualizzare il bucket `example_bucket` di Amazon S3. Questo esempio presuppone inoltre che tu stia eseguendo Windows AWS CLI su un computer che esegue Windows e che lo abbia già configurato AWS CLI con le tue credenziali. Per ulteriori informazioni, consultare la pagina relativa alla configurazione di AWS Command Line Interface.

La policy di attendibilità di esempio riportata di seguito è progettata per un'app per dispositivi mobili in cui l'utente accede tramite Amazon Cognito. In questo esempio, *us-east:12345678- ffff-ffff-ffff -123456* rappresenta l'ID del pool di identità assegnato da Amazon Cognito.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "RoleForCognito",
        "Effect": "Allow",
        "Principal": {"Federated": "cognito-identity.amazonaws.com"},
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-
east:12345678-ffff-ffff-ffff-123456"}}
    }
}
```

La policy delle autorizzazioni seguente consente agli utenti che assumono il ruolo di eseguire solo l'operazione `ListBucket` sul bucket `example_bucket` di Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

Per creare questo ruolo `Test-Cognito-Role`, è prima necessario salvare la policy di attendibilità precedente con il nome `trustpolicyforcognitofederation.json` e la policy di autorizzazione precedente con il nome `permspolicyforcognitofederation.json` nella cartella `policies` dell'unità `C:` locale. È quindi possibile utilizzare i comandi seguenti per creare il ruolo e collegare la policy inline.

```
# Create the role and attach the trust policy that enables users in an account to
 assume the role.
$ aws iam create-role --role-name Test-Cognito-Role --assume-role-policy-document
 file://C:\policies\trustpolicyforcognitofederation.json

# Attach the permissions policy to the role to specify what it is allowed to do.
```

```
aws iam put-role-policy --role-name Test-Cognito-Role --policy-name
 Perms-Policy-For-CognitoFederation --policy-document file://C:\policies
\permspolicyforcognitofederation.json
```

Creazione di un ruolo per AWS l'accesso federato (API)

Le procedure per creare un ruolo per il provider di identità supportato (OIDC o SAML) dalla AWS CLI sono identiche. La differenza consiste nel contenuto della policy di affidabilità creata in passaggi preliminari. Inizia seguendo le fasi descritte nella sezione dei prerequisiti per il tipo di provider in uso:

- Per un provider OIDC, consulta Prerequisiti per la creazione di un ruolo per OIDC.
- Per un provider SAML, consulta Prerequisiti per la creazione di un ruolo per SAML.

Creare un ruolo per la federazione delle identità (AWS API)

1. Crea un ruolo: CreateRole
2. Allega una politica di autorizzazioni al ruolo: AttachRolePolicy

   oppure

   Crea una politica di autorizzazioni in linea per il ruolo: PutRolePolicy
3. (Facoltativo) Aggiungi attributi personalizzati all'utente allegando tag: TagRole

   Per ulteriori informazioni, consulta Gestione dei tag sugli utenti IAM (AWS CLI o AWS API).
4. (Facoltativo) Imposta il limite delle autorizzazioni per il ruolo: PutRolePermissionsBoundary

   Il limite delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un ruolo. I limiti delle autorizzazioni sono una funzionalità avanzata. AWS

Creare un ruolo per la federazione OpenID Connect (console)

Puoi utilizzare i provider di identità federati OpenID Connect (OIDC) invece di creare utenti nel tuo. AWS Identity and Access Management Account AWS Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori informazioni sulla federazione e, vedere. IdPs Provider di identità e federazione

Prerequisiti per la creazione di un ruolo per OIDC

Prima di poter creare un ruolo per la federazione OIDC, è necessario completare i seguenti passaggi preliminari.

Per prepararsi a creare un ruolo per la federazione OIDC

1. Registrati con uno o più servizi che offrono l'identità OIDC federata. Se stai creando un'app che richiede l'accesso alle tue AWS risorse, configurala anche con le informazioni del provider. Al momento della registrazione, il gestore fornisce un ID applicazione o destinatario univoco per l'app. Provider diversi utilizzano una terminologia diversa per questo processo. Questa guida utilizza il termineconfigurare per il processo di identificazione dell'applicazione con il provider. È possibile configurare più app con ogni provider o più provider con una sola app. Consulta le informazioni sull'utilizzo degli IdP specificate di seguito:

   - Centro Sviluppatori di Login with Amazon
   - Aggiunta dell'accesso a Facebook a un'app o a un sito Web sul sito degli sviluppatori di Facebook.
   - Utilizzo di OAuth 2.0 per l'accesso (OpenID Connect) sul sito degli sviluppatori di Google.
2. Dopo aver ricevuto le informazioni richieste dall'IdP, crea un IdP in IAM. Per ulteriori informazioni, consulta Creare un provider di identità OpenID Connect (OIDC) in IAM.

   > **Important**
   >
   > Se utilizzi un IdP OIDC di Google, Facebook o Amazon Cognito, non occorre creare un IdP IAM separato nella AWS Management Console. Questi provider di identità OIDC sono già integrati AWS e possono essere utilizzati. Ignora questa fase e vai alla fase successiva per creare nuovi ruoli utilizzando l'IdP.
3. Prepara le policy per il ruolo che verrà assunto dagli utenti autenticati dal provider di identità. Come qualsiasi altro ruolo, anche il ruolo per un'app per dispositivi mobili include due policy. Una è la policy di affidabilità, che specifica chi può assumere il ruolo. L'altra è la policy di autorizzazione, che specifica le operazioni e le risorse AWS a cui l'app per dispositivi mobili può accedere o meno.

   Per il Web IdPs, ti consigliamo di utilizzare Amazon Cognito per gestire le identità. In tal caso, si utilizza una policy di attendibilità simile all'esempio seguente.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"Federated": "cognito-identity.amazonaws.com"},
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {
            "StringEquals": {"cognito-identity.amazonaws.com:aud": "us-
east-2:12345678-abcd-abcd-abcd-123456"},
            "ForAnyValue:StringLike": {"cognito-identity.amazonaws.com:amr":
 "unauthenticated"}
        }
    }
}
```

Sostituisci `us-east-2:12345678-abcd-abcd-abcd-123456` con l'ID del pool di identità che ti ha assegnato Amazon Cognito.

Se configuri manualmente un IdP OIDC, quando crei la policy di fiducia, devi utilizzare tre valori che assicurino che solo la tua app possa assumere il ruolo:

- Per l'elemento `Action`, si utilizza l'operazione `sts:AssumeRoleWithWebIdentity`.
- Per l'elemento `Principal`, usa la stringa `{"Federated":`*`providerUrl/providerArn`*`}`.
  - Per alcuni OIDC comuni IdPs, è un URL. *`providerUrl`* Gli esempi seguenti includono metodi per specificare il principale per alcuni casi comuni: IdPs

    ```
    "Principal":{"Federated":"cognito-identity.amazonaws.com"}
    ```

    ```
    "Principal":{"Federated":"www.amazon.com"}
    ```

    ```
    "Principal":{"Federated":"graph.facebook.com"}
    ```

    ```
    "Principal":{"Federated":"accounts.google.com"}
    ```
  - Per gli altri gestori OIDC, utilizza il nome della risorsa Amazon (ARN) dell'IdP OIDC creato in Step 2, come nell'esempio seguente:

    ```
    "Principal":{"Federated":"arn:aws:iam::123456789012:oidc-provider/
    server.example.com"}
    ```

- Per l'elemento `Condition`, si utilizza una condizione `StringEquals` per limitare le autorizzazioni. È necessario testare l'ID del pool di identità per Amazon Cognito o l'ID app per altri provider. L'ID del pool di identità dovrebbe corrispondere all'ID app che hai ricevuto durante la configurazione dell'app con l'IdP. Questa corrispondenza tra gli ID assicura che la richiesta provenga dalla tua app.

  > **Note**
  >
  > I ruoli IAM per i pool di identità di Amazon Cognito si affidano al responsabile del servizio `cognito-identity.amazonaws.com` per assumere il ruolo. I ruoli di questo tipo devono contenere almeno una chiave di condizione per limitare i responsabili che possono assumere il ruolo.
  > Considerazioni aggiuntive si applicano ai pool di identità di Amazon Cognito che presuppongono ruoli IAM su più account. Le politiche di fiducia di questi ruoli devono accettare il principio del `cognito-identity.amazonaws.com` servizio e devono contenere la chiave di `aud` condizione per limitare l'assunzione di ruoli agli utenti dei pool di identità previsti. Una policy che si fida dei pool di identità di Amazon Cognito senza questa condizione comporta il rischio che un utente proveniente da un pool di identità non intenzionale possa assumere il ruolo. Per ulteriori informazioni, consulta le politiche di fiducia per i ruoli IAM nell'autenticazione Basic (Classic) nella Amazon Cognito Developer Guide.

  Crea un elemento condizione simile agli esempi seguenti, a seconda dell'IdP in uso:

  ```
  "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud":
  "us-east:12345678-ffff-ffff-ffff-123456"}}
  ```

  ```
  "Condition": {"StringEquals": {"www.amazon.com:app_id":
  "amzn1.application-oa2-123456"}}
  ```

  ```
  "Condition": {"StringEquals": {"graph.facebook.com:app_id":
  "111222333444555"}}
  ```

  ```
  "Condition": {"StringEquals": {"accounts.google.com:aud":
  "666777888999000pro0"}}
  ```

Per i provider OIDC, si utilizza l'URL completo del provider di identità OIDC con la chiave di contesto aud, come nell'esempio seguente:

```
"Condition": {"StringEquals": {"server.example.com:aud":
"appid_from_oidc_idp"}}
```

> **Note**
>
> I valori per il principale nella policy di attendibilità per il ruolo sono specifici dell'IdP. Un ruolo per OIDC può specificare solo un principale. Pertanto, se l'app per dispositivi mobili consente agli utenti di effettuare l'accesso da più di un IdP, devi creare un ruolo separato per ogni IdP da supportare. Crea policy di attendibilità separate per ogni IdP.

Se un utente utilizza un'app per dispositivi mobili per accedere da Login with Amazon, si applica la policy di attendibilità di esempio riportata di seguito. Nell'esempio, *amzn1.application-oa2-123456* rappresenta l'ID app che Amazon ha assegnato al momento della configurazione dell'app con Login with Amazon.

```
{
      "Version": "2012-10-17",
      "Statement": [{
          "Sid": "RoleForLoginWithAmazon",
          "Effect": "Allow",
          "Principal": {"Federated": "www.amazon.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"www.amazon.com:app_id":
 "amzn1.application-oa2-123456"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Facebook, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *111222333444555* rappresenta l'ID app assegnato da Facebook.

```
{
      "Version": "2012-10-17",
```

```
      "Statement": [{
          "Sid": "RoleForFacebook",
          "Effect": "Allow",
          "Principal": {"Federated": "graph.facebook.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"graph.facebook.com:app_id":
 "111222333444555"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Google, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *666777888999000* rappresenta l'ID app assegnato da Google.

```
{
      "Version": "2012-10-17",
      "Statement": [{
          "Sid": "RoleForGoogle",
          "Effect": "Allow",
          "Principal": {"Federated": "accounts.google.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
          "Condition": {"StringEquals": {"accounts.google.com:aud":
 "666777888999000"}}
      }]
  }
```

Se un utente utilizza un'app per dispositivi mobili per accedere da Amazon Cognito, si applica la policy di attendibilità di esempio riportata di seguito. In questo esempio, *us-east:12345678-ffff-ffff-ffff -123456 rappresenta l'ID del pool di identità assegnato da Amazon Cognito*.

```
{
      "Version": "2012-10-17",
      "Statement": [{
          "Sid": "RoleForCognito",
          "Effect": "Allow",
          "Principal": {"Federated": "cognito-identity.amazonaws.com"},
          "Action": "sts:AssumeRoleWithWebIdentity",
```

```
          "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-
east:12345678-ffff-ffff-ffff-123456"}}
      }]
  }
```

Creazione di un ruolo per OIDC

Una volta completati i prerequisiti, puoi creare il ruolo in IAM. La procedura seguente descrive come creare il ruolo per la federazione OIDC in. AWS Management Console Per creare un ruolo dall' AWS API AWS CLI or, consulta le procedure all'indirizzo. Creazione di un ruolo per un provider di identità di terza parte (federazione)

> **Important**
>
> Se utilizzi Amazon Cognito, utilizza la console di Amazon Cognito per configurare i ruoli. Altrimenti, usa la console IAM per creare un ruolo per la federazione OIDC.

Per creare un ruolo IAM per la federazione OIDC

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.
3. Scegli il tipo di ruolo OIDC.
4. Per Identity provider (Gestore dell'identità digitale [IdP]), scegli l'IdP per il ruolo:
   - Se vuoi creare un ruolo per un singolo IdP Web, scegli Login with Amazon, Facebook o Google.

     > **Note**
     >
     > Devi creare un ruolo separato per ogni IdP che intendi supportare.

   - Se vuoi creare un ruolo per uno scenario avanzato per Amazon Cognito, scegli Amazon Cognito.

> **Note**
>
> Devi creare manualmente un ruolo da utilizzare con Amazon Cognito solo quando operi in uno scenario avanzato. In caso contrario, i ruoli possono essere creati da Amazon Cognito. Per ulteriori informazioni su Amazon Cognito, consulta la pagina Provider di identità esterni di pool di identità (identità federate) nella Guida per gli sviluppatori di Amazon Cognito.

- Se desideri creare un ruolo per GitHub Actions, devi iniziare aggiungendo il provider GitHub OIDC a IAM. Dopo aver aggiunto il provider GitHub OIDC a IAM, scegli token.actions.githubusercontent.com.

> **Note**
>
> Per informazioni su come configurare AWS l'OIDC GitHub di To Trust come identità federata, consulta GitHub Docs - Configuring OpenID Connect in Amazon Web Services. Per informazioni sulle best practice per limitare l'accesso ai ruoli associati a IAM IdP GitHub for, Configurazione di un ruolo per il provider di GitHub identità OIDC consulta questa pagina.

5. Inserisci l'identificatore per l'applicazione. L'etichetta relativa all'identificatore cambia in base al gestore scelto:

   - Se vuoi creare un ruolo per Login with Amazon, inserisci l'ID app nella casella Application ID (ID applicazione).
   - Se vuoi creare un ruolo per Facebook, inserisci l'ID app nella casella Application ID (ID applicazione).
   - Se vuoi creare un ruolo per Google, inserisci il nome del destinatario nella casella Audience (Destinatario).
   - Se vuoi creare un ruolo per Amazon Cognito, inserisci l'ID del pool di identità che hai creato per le applicazioni Amazon Cognito nella casella Identity Pool ID (ID pool di identità).
   - Se desideri creare un ruolo per GitHub Actions, inserisci i seguenti dettagli:
     - Per Pubblico, scegli `sts.amazonaws.com`.
     - Per GitHub l'organizzazione, inserisci il nome GitHub dell'organizzazione. Il nome GitHub dell'organizzazione è obbligatorio e deve essere alfanumerico, compresi i trattini (-). Non puoi usare caratteri jolly (* e?) nel nome dell' GitHub organizzazione.

- (Facoltativo) Per il GitHub repository, inserisci il nome del GitHub repository. Se non specifichi un valore, viene utilizzato un carattere jolly (*) per impostazione predefinita.
- (Facoltativo) Per il GitHub ramo, inserisci il nome del GitHub ramo. Se non specifichi un valore, viene utilizzato un carattere jolly (*) per impostazione predefinita.

6. (Facoltativo) In Condizione (facoltativo) scegli Aggiungi condizioneper creare condizioni aggiuntive che devono essere soddisfatte prima che gli utenti dell'applicazione possano utilizzare le autorizzazioni concesse dal ruolo. Ad esempio, puoi aggiungere una condizione che conceda l'accesso alle AWS risorse solo per uno specifico ID utente IAM. Puoi anche aggiungere condizioni alla policy di attendibilità dopo la creazione del ruolo. Per ulteriori informazioni, consulta Modifica di una policy di attendibilità del ruolo (Console).
7. Controlla le informazioni OIDC e poi scegli Avanti.
8. IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Seleziona la policy delle autorizzazioni da utilizzare o scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Seleziona la casella di controllo accanto alle politiche di autorizzazione che desideri vengano applicate agli utenti OIDC. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.
9. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

   Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.
10. Seleziona Successivo.
11. In Role name, (Nome ruolo), inserisci un nome. I nomi dei ruoli devono essere univoci all'interno del tuo. Account AWS Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** sia **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.
12. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
13. Per modificare i casi d'uso e le autorizzazioni per il ruolo, scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni).
14. (Facoltativo) Per aggiungere metadati al ruolo, collegare i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.

15. Rivedere il ruolo e scegliere Crea ruolo.

Configurazione di un ruolo per il provider di GitHub identità OIDC

Se si utilizza GitHub come provider di identità (IdP) OpenID Connect (OIDC), la best practice consiste nel limitare le entità che possono assumere il ruolo associato all'IDP IAM. Quando includi una dichiarazione di condizione nella policy di fiducia, puoi limitare il ruolo a un' GitHuborganizzazione, un repository o una filiale specifici. Puoi usare la chiave di condizione `token.actions.githubusercontent.com:sub` con operatori di condizione delle stringhe per limitare l'accesso. Ti consigliamo di limitare la condizione a un insieme specifico di repository o filiali all'interno dell'organizzazione GitHub . Per informazioni su come configurare AWS l'OIDC GitHub di To Trust come identità federata, consulta GitHub Docs - Configuring OpenID Connect in Amazon Web Services.

Se utilizzi GitHub ambienti in flussi di lavoro operativi o in policy OIDC, ti consigliamo vivamente di aggiungere regole di protezione all'ambiente per una maggiore sicurezza. Utilizza i rami e i tag di distribuzione per limitare i rami e i tag che possono essere distribuiti nell'ambiente. Per ulteriori informazioni sulla configurazione degli ambienti con regole di protezione, consulta i rami e i tag di distribuzione nell'articolo Utilizzo GitHub degli ambienti per la distribuzione.

Quando GitHub OIDC IdP è il Principal affidabile per il tuo ruolo, IAM verifica la condizione della policy di fiducia del ruolo per verificare che la chiave della condizione `token.actions.githubusercontent.com:sub` sia presente e che il suo valore non sia solo un carattere jolly (* e?) o null. IAM esegue questo controllo quando la policy di attendibilità viene creata o aggiornata. Se la chiave di condizione `token.actions.githubusercontent.com:sub` non è presente o il valore della chiave non soddisfa i criteri di valore indicati, la richiesta avrà esito negativo e restituirà un errore.

> **Important**
>
> Se non limiti la chiave di condizione a un'organizzazione o `token.actions.githubusercontent.com:sub` a un repository specifici, GitHub le azioni di organizzazioni o repository al di fuori del tuo controllo possono assumere ruoli associati all' GitHub IdP IAM nel tuo account. AWS

L'esempio seguente di policy di fiducia limita l'accesso all' GitHub organizzazione, al repository e alla filiale definiti. Il `token.actions.githubusercontent.com:sub` valore della chiave di condizione nell'esempio seguente è il formato predefinito del valore dell'oggetto documentato da. GitHub

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::012345678910:oidc-provider/
token.actions.githubusercontent.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
          "token.actions.githubusercontent.com:sub":
 "repo:GitHubOrg/GitHubRepo:ref:refs/heads/GitHubBranch"
        }
      }
    }
  ]
}
```

La condizione di esempio seguente limita l'accesso all' GitHub organizzazione e all'archivio definiti, ma concede l'accesso a qualsiasi ramo all'interno del repository.

```
"Condition": {
  "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com"
  },
  "StringLike": {
    "token.actions.githubusercontent.com:sub": "repo:GitHubOrg/GitHubRepo:*"
  }
}
```

La condizione di esempio seguente limita l'accesso a qualsiasi repository o ramo all'interno dell'organizzazione definita. GitHub Si consiglia di limitare la chiave di condizione `token.actions.githubusercontent.com:sub` a un valore specifico che limiti l'accesso ad GitHub Actions dall'interno GitHub dell'organizzazione.

```
"Condition": {
  "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com"
  },
  "StringLike": {
    "token.actions.githubusercontent.com:sub": "repo:GitHubOrg/*"
  }
}
```

Per ulteriori informazioni sulle chiavi di federazione OIDC disponibili per il controllo delle condizioni nelle politiche, vedere. Chiavi disponibili per la federazione AWS OIDC

Creare un ruolo per la federazione SAML 2.0 (console)

Puoi utilizzare la federazione SAML 2.0 invece di creare utenti IAM nella tua. Account AWS Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori informazioni sulla federazione e sui provider di identità, consultare Provider di identità e federazione.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per i dettagli, consulta l'articolo del AWS Security Blog Come utilizzare gli endpoint SAML regionali per il failover.

Prerequisiti per la creazione di un ruolo per SAML

Prima di creare un ruolo per la federazione SAML 2.0, devi completare i seguenti passaggi obbligatori.

Preparazione per la creazione di un ruolo per la federazione SAML 2.0

1. Prima di creare un ruolo per la federazione basata su SAML, devi creare un provider SAML in IAM. Per ulteriori informazioni, consulta Crea un provider di identità SAML in IAM.
2. Prepara le policy per il ruolo che verrà assunto dagli utenti autenticati con SAML 2.0. Come qualsiasi altro ruolo, anche i ruoli per la federazione SAML includono due policy. Una è la policy di attendibilità del ruolo, che specifica chi può assumere il ruolo. L'altra è la politica di autorizzazione IAM che specifica le AWS azioni e le risorse a cui l'utente federato può o negato l'accesso.

Al momento della creazione della policy di attendibilità per il ruolo, devi utilizzare tre valori che garantiscono che solo la tua applicazione possa assumere il ruolo:

- Per l'elemento `Action`, si utilizza l'operazione `sts:AssumeRoleWithSAML`.
- Per l'elemento `Principal`, usa la stringa `{"Federated":`*`ARNofIdentityProvider`*`}`. Sostituire *`ARNofIdentityProvider`* con l'ARN del provider di identità SAML creato in Step 1.
- Per l'elemento `Condition`, utilizzare una condizione `StringEquals` per verificare che l'attributo `saml:aud` della risposta SAML corrisponda all'endpoint della federazione SAML per AWS.

L'esempio seguente mostra una policy di affidabilità concepita per un utente federato SAML:

```
{
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Action": "sts:AssumeRoleWithSAML",
      "Principal": {"Federated": "arn:aws:iam::account-id:saml-provider/PROVIDER-
NAME"},
      "Condition": {"StringEquals": {"SAML:aud": "https://signin.aws.amazon.com/
saml"}}
    }
  }
```

Sostituisci l'ARN principale con l'ARN effettivo del provider SAML, creato in IAM. L'ARN include l'ID account e il nome del provider.

Creazione di un ruolo per SAML

Dopo aver completato i passaggi dei prerequisiti, è possibile creare il ruolo per la federazione basata su SAML.

Per creare un ruolo per una federazione basata su SAML

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegli Ruoli e quindi Crea ruolo.

3. Selezionare il tipo di ruolo SAML 2.0 federation (Federazione SAML 2.0).
4. In Select a SAML provider (Seleziona un gestore dell'identità digitale SAML), scegli il gestore per il ruolo.
5. Selezionare il metodo di livello di accesso SAML 2.0.

   - Scegli Consenti solo l'accesso programmatico per creare un ruolo che può essere assunto a livello di codice dall' AWS API oppure. AWS CLI
   - Scegli Consenti AWS Management Console accesso e programmazione per creare un ruolo che può essere assunto a livello di codice e da. AWS Management Console

   I due comandi sono simili, ma il ruolo che può essere assunto anche tramite console include una policy di affidabilità con una condizione particolare. Tale condizione verifica in modo esplicito che il destinatario SAML (attributo `SAML:aud`) sia impostato sull'endpoint di accesso di AWS per SAML (https://signin.aws.amazon.com/saml).
6. Se si sta creando un ruolo per l'accesso programmatico, scegliere un attributo dall'elenco Attributo. Dopodiché, nella casella Value (Valore), inserisci un valore da includere nel ruolo. In questo modo, l'accesso al ruolo viene limitato agli utenti dal provider di identità la cui risposta di autenticazione SAML (asserzione) includa gli attributi specificati. Per fare in modo che il ruolo sia limitato a un sottoinsieme di utenti all'interno dell'organizzazione, specificare almeno un attributo.

   Se si sta creando un ruolo per l'accesso programmatico e dalla console, l'attributo `SAML:aud` viene aggiunto automaticamente e impostato sull'URL dell'endpoint SAML di AWS (https://signin.aws.amazon.com/saml).
7. Per aggiungere ulteriori condizioni relative agli attributi alla policy di attendibilità, scegli Condition (optional) (Condizione [facoltativo]), seleziona la condizione aggiuntiva e specifica un valore.

   > **Note**
   >
   > L'elenco include gli attributi SAML più comunemente utilizzati. IAM supporta attributi aggiuntivi che puoi usare per creare condizioni. Per un elenco degli attributi supportati, consulta Chiavi disponibili per la federazione SAML. Se si necessita di una condizione per un attributo SAML supportato che non è nell'elenco, è possibile aggiungere tale condizione manualmente. A tale scopo, modificare la policy di attendibilità dopo aver creato il ruolo.
8. Verifica le informazioni di attendibilità di SAML 2.0, quindi scegli Next (Successivo).

9. IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Seleziona la policy delle autorizzazioni da utilizzare o scegli Create policy (Crea policy) per aprire una nuova scheda del browser e creare una nuova policy da zero. Per ulteriori informazioni, consulta Creazione di policy IAM. Una volta creata la policy, chiudere la scheda e tornare alla scheda originale. Seleziona la casella di controllo accanto alle politiche di autorizzazione che desideri vengano applicate agli utenti federati OIDC. È anche possibile non selezionare le policy ora e collegarle al ruolo in un secondo momento. Per default, un ruolo non dispone di autorizzazioni.
10. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata.

    Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). Selezionare la policy da utilizzare per il limite delle autorizzazioni.
11. Seleziona Successivo.
12. Scegli Prossimo: Rivedi.
13. In Role name, (Nome ruolo), inserisci un nome. I nomi dei ruoli devono essere univoci all'interno del tuo. Account AWS Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.
14. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
15. Scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni) per modificare i casi d'uso e le autorizzazioni per il ruolo.
16. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
17. Rivedere il ruolo e scegliere Crea ruolo.

Una volta creato il ruolo, è possibile completare la relazione di trust SAML configurando il software provider di identità con informazioni su AWS. Queste informazioni includono i ruoli che devono utilizzare gli utenti federati. Tale operazione viene definita configurazione della relazione di trust fra IdP e AWS. Per ulteriori informazioni, consulta Configura il tuo IdP SAML 2.0 con la fiducia dei relying party e l'aggiunta di claim.

Creazione di un ruolo utilizzando policy di attendibilità personalizzate (console)

Puoi creare una politica di fiducia personalizzata per delegare l'accesso e consentire ad altri di eseguire azioni nel tuo Account AWS. Per ulteriori informazioni, consulta Creazione di policy IAM.

Per informazioni su come utilizzare i ruoli per delegare le autorizzazioni, consultare Termini e concetti dei ruoli.

Creazione di un ruolo IAM utilizzando una policy di attendibilità personalizzata (console)

Puoi utilizzare il AWS Management Console per creare un ruolo che un utente IAM può assumere. Ad esempio, supponiamo che l'organizzazione disponga di più Account AWS elementi per isolare un ambiente di sviluppo da un ambiente di produzione. Per informazioni di alto livello sulla creazione di un ruolo che consenta agli utenti nell'account di sviluppo di accedere alle risorse nell'account di produzione, consulta la sezione Esempio di uno scenario in cui si utilizzano account di sviluppo e produzione separati.

Creare un ruolo utilizzando una policy di attendibilità personalizzata (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, selezionare Roles (Ruoli) e Crea ruolo.
3. Scegli il tipo di ruolo Custom trust policy (Policy di attendibilità personalizzata).
4. Nella sezione Custom trust policy (Policy di attendibilità personalizzata), inserisci o incolla la policy di attendibilità personalizzata per il ruolo. Per ulteriori informazioni, consulta Creazione di policy IAM.
5. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).
6. Seleziona la casella di controllo accanto alla policy di attendibilità personalizzata che hai creato.
7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.

   Apri la sezione Permissions boundary (Limite delle autorizzazioni) e scegli Use a permissions boundary to control the maximum role permissions (Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo). IAM include un elenco delle politiche AWS gestite e gestite dai clienti nel tuo account. Selezionare la policy da utilizzare per il limite delle autorizzazioni.

8. Seleziona Successivo.
9. Il grado di personalizzazione per Nome ruolo viene definito dal servizio. Se il servizio definisce il nome del ruolo, questa opzione non può essere modificata. In altri casi, il servizio può definire un prefisso per il ruolo e consentire all'utente di aggiungere un suffisso opzionale. Alcuni servizi consentono di specificare l'intero nome del ruolo.

   Se possibile, inserisci un nome del ruolo o un suffisso del nome del ruolo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.
10. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
11. Scegli Edit (Modifica) nelle sezioni Step 1: Select trusted entities (Fase 1: seleziona le entità attendibili) o Step 2: Add permissions (Fase 2: aggiungi autorizzazioni) per modificare la policy personalizzata e le autorizzazioni per il ruolo.
12. (Facoltativo) Aggiungere metadati al ruolo collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
13. Rivedere il ruolo e scegliere Crea ruolo.

Esempi di policy per la delega dell'accesso

Gli esempi seguenti mostrano come è possibile consentire o concedere Account AWS l'accesso alle risorse di un altro Account AWS. Per ulteriori informazioni su come creare una policy IAM utilizzando questi documenti di policy JSON, consulta the section called “Creazione di policy utilizzando l'editor JSON”.

Argomenti

- Utilizzo dei ruoli per delegare l'accesso alle risorse di altre Account AWS risorse
- Utilizzo di una policy per delegare l'accesso ai servizi
- Utilizzo di una policy basata sulle risorse per delegare l'accesso a un bucket Amazon S3 in un altro account
- Utilizzo di una policy basata sulle risorse per delegare l'accesso a una coda Amazon SQS in un altro account
- Impossibile delegare l'accesso quando l'accesso all'account è rifiutato

Utilizzo dei ruoli per delegare l'accesso alle risorse di altre Account AWS risorse

Per un tutorial che mostra come utilizzare i ruoli IAM per concedere agli utenti di un account l'accesso alle risorse AWS che si trovano in un altro account, consulta Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.

> **Important**
>
> È possibile includere l'ARN per un ruolo o utente specifico nell'elemento `Principal` di una policy di affidabilità del ruolo. Quando si salva la policy, AWS trasforma l'ARN in un ID principale univoco. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo o l'utente. Questa ID nella console non è normalmente presente, in quanto c'è anche una trasformazione inversa verso il nome ARN quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina il ruolo o l'utente, la relazione viene interrotta. La policy non è più applicabile, anche se si ricrea l'utente o il ruolo perché non corrisponde all'ID principale archiviato nella policy di attendibilità. Quando ciò accade, l'ID principale viene visualizzato nella console perché non è più AWS possibile mapparlo su un ARN. Il risultato è che se si elimina e si ricrea un utente o un ruolo referenziato in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire il nome ARN. L'utente o il ruolo viene trasformato nel nuovo ID principale quando si salva la policy.

Utilizzo di una policy per delegare l'accesso ai servizi

L'esempio seguente mostra una policy che può essere collegata a un ruolo. La policy consente a due servizi, Amazon EMR e AWS Data Pipeline, di assumere il ruolo. I servizi possono eseguire qualsiasi attività concesse da una policy di autorizzazioni assegnata al ruolo (non visualizzato). Per specificare più principali del servizio, non si specificano due elementi `Service`, è possibile averne solo uno. Utilizzare invece una serie di principali del servizio come il valore di un elemento singolo `Service`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "elasticmapreduce.amazonaws.com",
```

```
          "datapipeline.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Utilizzo di una policy basata sulle risorse per delegare l'accesso a un bucket Amazon S3 in un altro account

In questo esempio, l'account A utilizza una policy basata sulle risorse (una policy del bucket Amazon S3) per concedere all'account B l'accesso completo al bucket S3 dell'account A. A questo punto, l'account B crea una policy utente IAM per delegare tale accesso al bucket dell'account A a uno degli utenti nell'account B.

La policy del bucket S3 nell'account A potrebbe essere simile alla policy seguente. In questo esempio, il bucket S3 dell'account A è denominato mybucket e il numero dell'account B è 111122223333. Non specifica alcun utente o gruppo nell'account B, ma solo l'account stesso.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AccountBAccess1",
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

In alternativa, l'account A può utilizzare le liste di controllo accessi (ACL) di Amazon S3 per concedere l'accesso all'account B a un bucket S3 o a un singolo oggetto all'interno di un bucket. In questo caso, l'unica cosa che cambia è il modo in cui l'account A concede l'accesso all'account B. L'account B utilizza ancora una policy per delegare l'accesso a un gruppo IAM nell'account B, come descritto nella parte successiva di questo esempio. Per maggiori informazioni sul controllo dell'accesso a bucket e oggetti S3, passa a Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service.

L'amministratore dell'account B potrebbe creare le seguenti policy di esempio. La policy permette l'accesso in lettura a un gruppo o un utente nell'account B. La policy precedente concede l'accesso all'account B. Tuttavia, i singoli gruppi e gli utenti dell'account B non possono accedere alla risorsa finché una policy utente o di gruppo non concede esplicitamente le autorizzazioni alla risorsa. Le autorizzazioni in questa policy possono essere solo un subset di quelli nella precedente policy multiaccount. L'account B non può concedere più autorizzazioni ai propri gruppi e utenti rispetto a quanti concessi dall'account A all'account B nella prima policy. In questa policy, l'elemento `Action` è esplicitamente definito per permettere solo operazioni `List` e l'elemento `Resource` di questa policy corrisponde all'elemento `Resource` per la policy del bucket implementata dall'account A.

Per implementare questa policy, l'account B utilizza IAM per collegarla all'utente (o al gruppo) appropriato nell'account B.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:List*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

Utilizzo di una policy basata sulle risorse per delegare l'accesso a una coda Amazon SQS in un altro account

Nell'esempio seguente, l'account A ha una coda Amazon SQS che utilizza una policy basata sulla risorsa collegata alla coda per concedere l'accesso in coda all'account B. Quindi, l'account B utilizza una policy di gruppo IAM per delegare l'accesso a un gruppo nell'account B.

La seguente policy di coda di esempio fornisce all'account B l'autorizzazione per eseguire le operazioni `SendMessage` e `ReceiveMessage` sulla coda dell'account A denominata queue1, ma solo tra mezzogiorno e le 15:00 del 30 novembre 2014. Il numero dell'account B è 1111-2222-3333. L'account A usa Amazon SQS per implementare questa policy.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
```

```
    "Principal": {"AWS": "111122223333"},
    "Action": [
      "sqs:SendMessage",
      "sqs:ReceiveMessage"
    ],
    "Resource": ["arn:aws:sqs:*:123456789012:queue1"],
    "Condition": {
      "DateGreaterThan": {"aws:CurrentTime": "2014-11-30T12:00Z"},
      "DateLessThan": {"aws:CurrentTime": "2014-11-30T15:00Z"}
    }
  }
}
```

La policy dell'account B per la delega dell'accesso a un gruppo nell'account B potrebbe essere simile all'esempio seguente. L'account B usa IAM per collegare questa policy a un gruppo (o utente).

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sqs:*",
    "Resource": "arn:aws:sqs:*:123456789012:queue1"
  }
}
```

Nell'esempio della policy utente IAM precedente, l'account B utilizza un carattere jolly per concedere all'utente l'accesso a tutte le operazioni Amazon SQS per la coda dell'account A. Tuttavia, l'account B può delegare l'accesso solo nella misura in cui all'account B è stato concesso l'accesso. Il gruppo dell'account B con la seconda policy può accedere alla coda solo tra mezzogiorno e le 15:00 del 30 novembre 2014. L'utente può eseguire solo le operazioni `SendMessage` e `ReceiveMessage`, come definito nella policy della coda Amazon SQS dell'account A.

Impossibile delegare l'accesso quando l'accesso all'account è rifiutato

An Account AWS non può delegare l'accesso alle risorse di un altro account se l'altro account ha negato esplicitamente l'accesso all'account principale dell'utente. Il rifiuto si propaga agli utenti di tale account indipendentemente dal fatto che gli utenti dispongano di policy esistenti che garantiscono loro l'accesso.

Ad esempio, l'account A scrive una policy bucket per il bucket S3 dell'account A che rifiuta esplicitamente l'accesso all'account B per il bucket dell'account A. Tuttavia, l'account B scrive una

policy utente IAM che concede a un utente dell'account B l'accesso al bucket dell'account A. Il rifiuto esplicito applicato al bucket S3 dell'account A si propaga agli utenti dell'account B e sostituisce la policy dell'utente IAM che concede l'accesso all'utente dell'account B. Per informazioni dettagliate su come vengono valutate le autorizzazioni, consulta Logica di valutazione delle policy.

La policy del bucket dell'account A potrebbe essere simile alla policy seguente. In questo esempio, il bucket S3 dell'account A è denominato mybucket e il numero dell'account B è 1111-2222-3333. L'account A usa Amazon S3 per implementare questa policy.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AccountBDeny",
    "Effect": "Deny",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::mybucket/*"
  }
}
```

Questo rifiuto esplicito sostituisce qualsiasi policy dell'account B che fornisce l'autorizzazione per accedere al bucket S3 nell'account A.

Utilizzo di ruoli IAM

Prima che un utente, applicazione o servizio possa utilizzare un ruolo che si è creato, è necessario concedere le autorizzazioni per passare al ruolo. È possibile utilizzare qualsiasi politica associata a gruppi o utenti per concedere le autorizzazioni necessarie. In questa sezione viene descritto come concedere agli utenti l'autorizzazione per l'utilizzo di un ruolo. Spiega inoltre come l'utente può passare a un ruolo dagli AWS Management Console Strumenti per Windows PowerShell, da AWS Command Line Interface (AWS CLI) e dall'`AssumeRole`API.

> **Important**
>
> Se crei un ruolo a livello programmatico anziché nella console IAM, hai l'opzione per aggiungere un `Path` con un massimo di 512 caratteri in aggiunta a `RoleName`, che può contenere fino a 64 caratteri. Tuttavia, se intendi utilizzare un ruolo con la funzione Cambia ruolo in AWS Management Console, la combinazione `Path` `RoleName` non può superare i 64 caratteri.

È possibile cambiare ruolo da AWS Management Console. Puoi assumere un ruolo chiamando un'operazione AWS CLI o API o utilizzando un URL personalizzato. Il metodo utilizzato determina chi può assumere il ruolo e per quanto tempo la sessione del ruolo della sessione può durare. Quando utilizzi `AssumeRole*` Operazioni API, il ruolo IAM assunto è la risorsa. L'utente o il ruolo che chiama le operazioni API `AssumeRole*` è il principale.

Confronto dei metodi per l'utilizzo di ruoli

Metodo per assumere il ruolo	Chi può assumere il ruolo	Metodo per specificare il ciclo di vita delle credenziali	Ciclo di vita delle credenziali (minimo | massimo | predefinito)
AWS Management Console	Utente (mediante lo scambio di ruoli)	Durata massima sessione nella pagina di riepilogo Ruolo	15 min | Impostazione durata massima sessione² | 1 ora
`assume-role` CLI oppure operazione API `AssumeRole`	Utente o ruolo¹	CLI `duration-seconds` oppure parametro API `DurationSeconds`	15 min | Impostazione durata massima sessione² | 1 ora
`assume-role-with-saml` CLI oppure operazione API `AssumeRoleWithSAML`	Tutti gli utenti autenticati utilizzando SAML	CLI `duration-seconds` oppure parametro API `DurationSeconds`	15 min | Impostazione durata massima sessione² | 1 ora
`assume-role-with-web-identity` CLI oppure operazione API	Qualsiasi utente autenticato tramite un provider OIDC	CLI `duration-seconds` oppure parametro API	15 min | Impostazione durata massima sessione² | 1 ora

Metodo per assumere il ruolo	Chi può assumere il ruolo	Metodo per specificare il ciclo di vita delle credenziali	Ciclo di vita delle credenziali (minimo | massimo | predefinito)
`AssumeRoleWithWebIdentity`		`DurationSeconds`	
Console URL creata con `AssumeRole`	Utente o ruolo	Parametro HTML `SessionDuration` nell'URL	15 min | 12 ore | 1 ora
Console URL creata con `AssumeRoleWithSAML`	Tutti gli utenti autenticati utilizzando SAML	Parametro HTML `SessionDuration` nell'URL	15 min | 12 ore | 1 ora
Console URL creata con `AssumeRoleWithWebIdentity`	Qualsiasi utente autenticato tramite un provider OIDC	Parametro HTML `SessionDuration` nell'URL	15 min | 12 ore | 1 ora

[1] L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, le nuove credenziali sono limitate a una durata massima di un'ora. Quando si utilizzano i ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2, tali applicazioni non sono soggette a questa limitazione.

[2] Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Per informazioni sulla modifica dell'impostazione della durata massima della sessione, consulta Modifica di un ruolo. Questa impostazione determina la durata massima della sessione che è possibile richiedere quando si ottengono le credenziali del ruolo. Ad esempio, quando si utilizzano le operazioni dell'AssumeRoleAPI* per assumere un ruolo, è possibile specificare la durata della sessione utilizzando il parametro. `DurationSeconds` Utilizzare questo parametro per specificare la durata

della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Agli utenti IAM che cambiano ruoli nella console viene concessa la durata massima della sessione o il tempo rimanente nella sessione dell'utente, a seconda di quale sia minore. Si supponga di impostare una durata massima di 5 ore su un ruolo. Un utente IAM che è stato collegato alla console per 10 ore (rispetto al valore massimo predefinito di 12) può cambiare ruolo. La durata della sessione di ruolo disponibile è di 2 ore. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo più avanti su questa pagina.

> **Note**
>
> - L'impostazione di durata massima delle sessioni non limita le sessioni assunte dai servizi AWS .
> - Le credenziali del ruolo IAM di Amazon EC2 non sono soggette alla durata massima delle sessioni configurata nel ruolo.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e aggiornano automaticamente tali credenziali. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di attendibilità dei ruoli per aggiungere il ruolo principale ARN o Account AWS ARN, vedere. Modifica di una policy di attendibilità del ruolo (Console)

Argomenti

- Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2
- Revoca delle credenziali di sicurezza temporanee per i ruoli IAM

Visualizzazione dell'impostazione di durata massima della sessione per un ruolo

È possibile specificare la durata massima della sessione per un ruolo utilizzando AWS Management Console o utilizzando l'API AWS CLI o AWS . Quando si utilizza un'operazione AWS CLI o API per assumere un ruolo, è possibile specificare un valore per il `DurationSeconds` parametro. Puoi utilizzare questo parametro per specificare la durata della sessione del ruolo, da 900 secondi (15 minuti) fino all'impostazione Durata massima della sessione CLI/API per il ruolo. Prima di specificare il parametro, è consigliabile visualizzare questa impostazione per il ruolo. Se si specifica un valore per il parametro `DurationSeconds` che è superiore all'impostazione massina, l'operazione ha esito negativo.

Per visualizzare la durata massima della sessione di un ruolo (console)

1. Nel pannello di navigazione della console IAM seleziona Ruoli.
2. Selezionare il nome del ruolo che si desidera visualizzare.
3. Accanto a Durata massima sessione, visualizza la durata massima della sessione concessa per il ruolo. Questa è la durata massima della sessione che puoi specificare nella tua AWS CLI operazione o API.

Per visualizzare l'impostazione della durata massima della sessione di un ruolo (AWS CLI)

1. Se non si conosce il nome del ruolo che si desidera assumere, eseguire il comando seguente per elencare i ruoli nell'account:
   - aws iam list-roles
2. Per visualizzare la durata massima della sessione del ruolo, eseguire il comando seguente. Visualizzare il parametro di durata massima della sessione.
   - aws iam get-role

Per visualizzare l'impostazione della durata massima della sessione (AWS API) di un ruolo

1. Se non si conosce il nome del ruolo che si desidera assumere, chiamare la seguente operazione per elencare i ruoli nell'account:

   - ListRoles

2. Per visualizzare la durata massima della sessione del ruolo, eseguire la seguente operazione. Visualizzare il parametro di durata massima della sessione.

   - GetRole

Concessione di autorizzazioni agli utenti per il cambio di ruoli

Quando un amministratore crea un ruolo per l'accesso multi-account, stabilisce l'attendibilità tra l'account proprietario del ruolo e le risorse (account che determina l'attendibilità) e l'account che contiene gli utenti (account attendibile). A tale scopo, l'amministratore dell'account attendibile specifica il numero dell'account attendibile come `Principal` nella policy di attendibilità del ruolo. Ciò consente potenzialmente a qualsiasi utente nell'account attendibile di assumere il ruolo. Per completare la configurazione, l'amministratore dell'account attendibile deve fornire a determinati gruppi o utenti dell'account l'autorizzazione per il passaggio al ruolo.

Concessione dell'autorizzazione per passare a un ruolo

1. In qualità di amministratore dell'account attendibile, crea una nuova policy per l'utente oppure modifica una policy esistente per aggiungere gli elementi richiesti. Per informazioni dettagliate, vedi Creazione o modifica della policy.
2. Quindi decidi come desideri eseguire la condivisione delle informazioni sul ruolo:

   - Role link (Link del ruolo): invia agli utenti un collegamento che indirizza alla pagina Switch Role (Cambia ruolo) con tutti i dettagli già compilati.
   - ID account o alias: fornisci a ciascun utente il nome del ruolo insieme al numero dell'ID account o all'alias dell'account. L'utente accede quindi alla pagina Switch Role (Cambia ruolo) e aggiunge i dettagli manualmente.

   Per informazioni dettagliate, vedi Fornire informazioni all'utente.

Tieni presente che puoi cambiare ruolo solo quando effettui l'accesso come utente IAM, come ruolo federato SAML o come ruolo con federazione delle identità Web. Non è possibile cambiare i ruoli se si effettua l'accesso come Utente root dell'account AWS.

> **Important**
>
> Non è possibile passare da un ruolo AWS Management Console a un ruolo che richiede un ExternalIdvalore. È possibile passare a tale ruolo solo chiamando l'API `AssumeRole` che supporta il parametro `ExternalId`.

> **Note**
>
> - In questo argomento sono descritte le policy per un utente poiché sostanzialmente concedi autorizzazioni a un utente per l'esecuzione di un'operazione. Tuttavia, consigliamo di non concedere le autorizzazioni direttamente a un singolo utente. Quando un utente assume un ruolo, gli vengono assegnate le autorizzazioni associate a quel ruolo.
> - Quando si cambia ruolo in AWS Management Console, la console utilizza sempre le credenziali originali per autorizzare il passaggio. Questo vale sia per l'accesso come utente IAM che come ruolo federato SAML o come ruolo federato di identità Web. Ad esempio, se passi al RuoloA, IAM; utilizza le tue credenziali utente originali o le credenziali del ruolo federato per determinare se è possibile assumere il RuoloA. Se provi quindi a passare al RuoloB mentre stai utilizzando il RuoloA, per autorizzare il tentativo vengono utilizzate le credenziali utente originali o le credenziali del ruolo federato. Le credenziali per RuoloA non vengono utilizzate per questa operazione.

Argomenti

Creazione o modifica della policy

Una policy che concede a un utente l'autorizzazione di assumere un ruolo deve includere una dichiarazione con effetto `Allow` per quanto segue:

- L'operazione `sts:AssumeRole`

- L'Amazon Resource Name (ARN) del ruolo in un elemento `Resource`

Agli utenti che ottengono la policy (mediante l'appartenenza a un gruppo o collegata direttamente) è consentito cambiare ruoli sulla risorsa specificata.

> **Note**
>
> Se `Resource` è impostato su `*`, l'utente può assumere qualsiasi ruolo in qualsiasi account che considera l'account utente attendibile. (In altre parole, la policy di attendibilità del ruolo specifica l'account dell'utente come `Principal`). Come best practice, si consiglia di seguire il principio di privilegio minimo e specificare l'ARN completo solo per i ruoli necessari per l'utente.

L'esempio seguente mostra una policy che consente all'utente di assumere ruoli in un unico account. Inoltre, la policy utilizza un carattere jolly (*) per specificare che l'utente può passare a un ruolo solo se il nome del ruolo inizia con le lettere `Test`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::account-id:role/Test*"
  }
}
```

> **Note**
>
> Le autorizzazioni che il ruolo concede all'utente non vengono aggiunte alle autorizzazioni già concesse per l'utente. Quando un utente passa a un ruolo, l'utente rinuncia temporaneamente alle proprie autorizzazioni originali in cambio di quelle concesse dal ruolo. Quando l'utente esce dal ruolo, le autorizzazioni originali dell'utente vengono ripristinate automaticamente. Ad esempio, assumiamo che le autorizzazioni dell'utente permettano di utilizzare le istanze Amazon EC2, ma le policy di autorizzazione del ruolo non concedano tali autorizzazioni. In tal caso, durante l'utilizzo del ruolo, l'utente non potrà utilizzare le istanze

Amazon EC2 nella console. Inoltre, le credenziali temporanee ottenute tramite `AssumeRole` non funzioneranno con le istanze Amazon EC2 in modo programmatico.

Fornire informazioni all'utente

Dopo aver creato un ruolo e concesso all'utente le autorizzazioni per passare a tale ruolo, è necessario fornire all'utente quanto segue:

- Il nome del ruolo
- L'ID o l'alias dell'account che contiene il ruolo

Puoi semplificare le operazioni per gli utenti inviando loro un collegamento preconfigurato con l'ID account e il nome del ruolo. Puoi visualizzare il collegamento al ruolo dopo aver completato la procedura guidata Crea ruolo selezionando il banner Visualizza ruolo o nella pagina Riepilogo del ruolo per qualsiasi ruolo abilitato per più account.

È inoltre possibile utilizzare il formato seguente per creare manualmente il collegamento. Sostituisci l'ID account o alias e il nome del ruolo per i due parametri nel seguente esempio:

```
https://signin.aws.amazon.com/switchrole?
account=your_account_ID_or_alias&roleName=optional_path/role_name
```

Consigliamo di suggerire agli utenti di consultare l'argomento Cambio di un ruolo (console) per guidarli nel processo. Per risolvere i problemi più comuni che si possono verificare quando si assume un ruolo, consulta la pagina Non è possibile assumere un ruolo.

Considerazioni

- Se crei il ruolo in maniera programmatica, puoi crearlo con un percorso e un nome. In tal caso, è necessario fornire il percorso completo e il nome del ruolo agli utenti in modo che possano specificare queste informazioni sulla pagina Cambia ruolo della AWS Management Console. Ad esempio: `division_abc/subdivision_efg/role_XYZ`.
- Se crei il ruolo in maniera programmatica, potrai aggiungere un `Path` con un massimo di 512 caratteri e un `RoleName`. Il nome del ruolo può contenere un massimo di 64 caratteri. Tuttavia, per utilizzare un ruolo con la funzione Switch Role in AWS Management Console, la combinazione `Path` e `RoleName` non può superare i 64 caratteri.

- Per motivi di sicurezza, puoi esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in AWS. È possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`. Puoi utilizzare `sts:RoleSessionName` anche per richiedere agli utenti di specificare un nome di sessione quando assumono un ruolo. Ciò consente di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi.

Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS

Per configurare molti AWS servizi, devi passare un ruolo IAM al servizio. Ciò consente al servizio di assumere successivamente il ruolo ed eseguire operazioni per tuo conto. Per la maggior parte dei servizi, è sufficiente passare il ruolo al servizio una sola volta durante la configurazione e non ogni volta che il servizio assume il ruolo. Ad esempio, si supponga di disporre di un'applicazione in esecuzione in un'istanza Amazon EC2. Tale applicazione richiede credenziali temporanee per l'autenticazione e autorizzazioni per autorizzare l'applicazione a eseguire operazioni in AWS. Quando configuri l'applicazione, devi passare un ruolo ad Amazon EC2 per l'utilizzo con l'istanza che fornisce tali credenziali. È possibile definire le autorizzazioni per le applicazioni in esecuzione nell'istanza collegando una policy IAM al ruolo. L'applicazione assume il ruolo ogni volta che è necessario per eseguire le operazioni consentite dal ruolo.

Per passare un ruolo (e le relative autorizzazioni) a un AWS servizio, un utente deve disporre delle autorizzazioni per passare il ruolo al servizio. Ciò consente agli amministratori di garantire che solo gli utenti autorizzati possano configurare un servizio con un ruolo che concede le autorizzazioni. Per consentire a un utente di passare un ruolo a un AWS servizio, devi concedere l'`PassRole`autorizzazione all'utente, al ruolo o al gruppo IAM dell'utente.

> ⚠ Warning
>
> - Puoi utilizzare l'`PassRole`autorizzazione solo per passare un ruolo IAM a un servizio che condivide lo stesso AWS account. Per passare un ruolo nell'Account A a un servizio nell'Account B, devi prima creare un ruolo IAM nell'Account B che possa assumere il ruolo dall'Account A, quindi il ruolo nell'Account B può essere passato al servizio. Per informazioni dettagliate, vedi Accesso alle risorse multi-account in IAM.

- Non cercare di controllare chi può passare un ruolo assegnando tag al ruolo e utilizzando la chiave di condizione `ResourceTag` in una policy con l'operazione `iam:PassRole`. Questo approccio non produce risultati affidabili.

Quando imposti l'`PassRole`autorizzazione, devi assicurarti che a un utente non venga assegnato un ruolo in cui il ruolo dispone di più autorizzazioni di quelle che desideri che l'utente abbia. Ad esempio, Alice potrebbe non essere autorizzata a eseguire alcuna azione su Amazon S3. Se Alice potesse trasferire un ruolo a un servizio che consente le azioni di Amazon S3, il servizio potrebbe eseguire azioni Amazon S3 per conto di Alice durante l'esecuzione del job.

Quando si specifica un ruolo collegato ai servizi, è necessario disporre anche delle autorizzazioni per inoltrare tale ruolo al servizio. Alcuni servizi creano automaticamente un ruolo collegato ai servizi nell'account quando si esegue un'azione in quel servizio. Ad esempio, Amazon EC2 Auto Scaling crea il ruolo collegato ai servizi `AWSServiceRoleForAutoScaling` la prima volta che crei un gruppo Auto Scaling. Se provi a specificare il ruolo collegato ai servizi quando crei un gruppo con scalabilità automatica senza l'autorizzazione `iam:PassRole`, viene visualizzato un messaggio di errore. Se non specifichi esplicitamente il ruolo, l'autorizzazione `iam:PassRole` non è richiesta e l'impostazione predefinita prevede l'utilizzo del ruolo `AWSServiceRoleForAutoScaling` per tutte le operazioni eseguite su quel gruppo. Per scoprire i servizi che supportano i ruoli collegati ai servizi, consulta AWS servizi che funzionano con IAM. Per scoprire quali servizi creano automaticamente un ruolo collegato ai servizi quando si esegue un'operazione in quel servizio, selezionare il collegamento Yes (Sì) e visualizzare il ruolo collegato ai servizi per il servizio.

Un utente può passare un ruolo ARN come parametro in qualsiasi operazione API che utilizza il ruolo per assegnare le autorizzazioni al servizio. Il servizio quindi verifica se l'utente dispone dell'autorizzazione `iam:PassRole`. Per limitare l'utente a passare solo i ruoli approvati, puoi filtrare l'autorizzazione `iam:PassRole` con l'elemento `Resources` dell'istruzione della policy IAM.

Puoi utilizzare l'`Condition`elemento in una policy JSON per testare il valore delle chiavi incluse nel contesto di richiesta di tutte le richieste. AWS Per ulteriori informazioni sull'utilizzo delle chiavi di condizione in una policy, consulta Elementi delle policy JSON IAM: Condition. La chiave di condizione `iam:PassedToService` può essere utilizzata per specificare il principale del servizio del servizio a cui è possibile passare un ruolo. Per ulteriori informazioni sull'utilizzo della chiave `iam:PassedToService` condition in una politica, consulta iam: PassedToService.

Esempio 1

Si immagini di voler concedere a un utente la possibilità di trasferire uno qualsiasi dei set di ruoli approvati al servizio Amazon EC2 all'avvio di un'istanza. È necessario disporre di tre elementi:

- Una policy di autorizzazioni IAM collegata al ruolo che determina quali operazioni può compiere il ruolo. Definire l'ambito delle autorizzazioni in modo da includere solo le operazioni che il ruolo deve effettuare e sole le risorse necessarie per tali operazioni. Puoi utilizzare una politica di autorizzazioni IAM AWS gestita o creata dal cliente.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [ "A list of the permissions the role is allowed to use" ],
        "Resource": [ "A list of the resources the role is allowed to access" ]
    }
}
```

- Una policy di attendibilità per il ruolo che consente al servizio di assumere tale ruolo. Ad esempio, è possibile collegare la seguente policy di affidabilità al ruolo con l'operazione `UpdateAssumeRolePolicy`. Questa policy di attendibilità consente ad Amazon EC2 di utilizzare il ruolo e le autorizzazioni associate al ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "TrustPolicyStatementThatAllowsEC2ServiceToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": { "Service": "ec2.amazonaws.com" },
       "Action": "sts:AssumeRole"
    }
}
```

- Una policy di autorizzazioni IAM collegata all'utente IAM che consente all'utente di trasferire solo i ruoli approvati. In genere si aggiunge `iam:GetRole` a `iam:PassRole` in modo che l'utente possa ottenere i dettagli del ruolo da passare. In questo esempio, l'utente può passare solo i ruoli esistenti nell'account specificato con nomi che iniziano con `EC2-roles-for-XYZ-`:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
```

```
        "Action": [
            "iam:GetRole",
            "iam:PassRole"
        ],
        "Resource": "arn:aws:iam::account-id:role/EC2-roles-for-XYZ-*"
    }]
}
```

Ora l'utente può avviare un'istanza Amazon EC2 con un ruolo assegnato. Le applicazioni in esecuzione nell'istanza possono accedere alle credenziali temporanee per il ruolo tramite i metadati del profilo dell'istanza. Le policy delle autorizzazioni collegate al ruolo determinano cosa può fare l'istanza.

Esempio 2

Amazon Relational Database Service (Amazon RDS) supporta una funzione chiamata Monitoraggio avanzato. Questa funzione consente ad Amazon RDS di monitorare un'istanza di database tramite un agente. Consente inoltre ad Amazon RDS di registrare i parametri su Amazon CloudWatch Logs. Per abilitare questa funzione, è necessario creare un ruolo di servizio per fornire le autorizzazioni Amazon RDS per monitorare e scrivere i parametri per i log.

Come creare un ruolo IAM per il monitoraggio avanzato di Amazon RDS

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Roles (Ruoli), quindi selezionare Create role (Crea ruolo).
3. Scegli il tipo di ruolo di AWS servizio, quindi in Casi d'uso per altri Servizi AWS, scegli il servizio RDS. Scegli RDS - Enhanced Monitoring (RDS - Monitoraggio avanzato), quindi seleziona Next (Successivo).
4. Scegli la politica di autorizzazione di AmazonRDS EnhancedMonitoringRole.
5. Seleziona Successivo.
6. In Role name (Nome ruolo), inserisci un nome del ruolo che consenta di identificarne lo scopo. I nomi dei ruoli devono essere univoci all'interno del tuo. Account AWS Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole. Quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante la procedura di accesso, il nome del ruolo non fa distinzione tra maiuscole e

minuscole. Poiché varie entità possono fare riferimento al ruolo, non puoi modificare il nome del ruolo dopo averlo creato.

7. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
8. (Facoltativo) Aggiungi metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
9. Rivedere il ruolo e scegliere Crea ruolo.

Il ruolo ottiene automaticamente una policy di affidabilità che concede le autorizzazioni del servizio `monitoring.rds.amazonaws.com` per assumere il ruolo. Dopo l'avvio, Amazon RDS potrà eseguire tutte le operazioni consentite dalla policy `AmazonRDSEnhancedMonitoringRole`.

L'utente che desideri possa accedere al monitoraggio avanzato necessita di una policy che includa un'istruzione che consenta all'utente di elencare i ruoli RDS e l'istruzione che consente di passare il ruolo, come nell'esempio seguente. Utilizza il tuo numero di account e sostituisci il nome del ruolo con il nome fornito nel passaggio 6.

```
{
  "Sid": "PolicyStatementToAllowUserToListRoles",
  "Effect": "Allow",
  "Action": ["iam:ListRoles"],
  "Resource": "*"
},
{
    "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",
    "Effect": "Allow",
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam::account-id:role/RDS-Monitoring-Role"
}
```

È possibile combinare questa istruzione con dichiarazioni in un'altra policy o collocarla nella policy personalizzata. Invece, per specificare che l'utente può passare qualsiasi ruolo che inizia con `RDS-`, puoi sostituire il nome del ruolo nella risorsa ARN con un carattere jolly, come nell'esempio seguente.

```
    "Resource": "arn:aws:iam::account-id:role/RDS-*"
```

Operazioni **`iam:PassRole`** nei log AWS CloudTrail

`PassRole`non è una chiamata API. `PassRole`è un'autorizzazione, il che significa che non vengono generati CloudTrail log per IAM`PassRole`. Per verificare quali ruoli vengono passati a quali Servizi AWS in CloudTrail, è necessario esaminare il CloudTrail registro che ha creato o modificato la AWS risorsa che riceve il ruolo. Ad esempio, un ruolo viene passato a una AWS Lambda funzione al momento della creazione. Il log per l'operazione `CreateFunction` mostra un record del ruolo passato alla funzione.

Cambio di un ruolo (console)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse AWS necessarie. In questo senso, è simile a un utente in AWS Identity and Access Management (IAM). Quando effettui l'accesso come utente, ottieni uno specifico set di autorizzazioni. Tuttavia, non accedi a un ruolo, ma una volta effettuato l'accesso puoi passare a un ruolo. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM.

> **Important**
>
> Le autorizzazioni dell'utente e di qualsiasi ruolo a cui si passa non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando passi a un ruolo, lasci temporaneamente le autorizzazioni utente e utilizzi le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Quando cambi ruolo in AWS Management Console, la console utilizza sempre le tue credenziali originali per autorizzare lo switch. Questo vale per l'accesso come utente IAM, come utente del Centro identità IAM, come ruolo federato SAML o come ruolo con federazione delle identità Web. Ad esempio, se passi al RuoloA, IAM utilizza le tue credenziali utente originali o le credenziali del ruolo federato per determinare se è possibile assumere il RuoloA. Se poi passi al ruolo B mentre utilizzi RolEA, utilizza comunque le credenziali dell'utente originale o del ruolo federato per autorizzare lo switch AWS , non le credenziali per RolEA.

Informazioni sul cambio dei ruoli nella console

In questa sezione sono fornite ulteriori informazioni sull'utilizzo della console IAM per cambiare ruolo.

> **Note:**
>
> - Non è possibile cambiare ruolo se si accede come. Utente root dell'account AWS Puoi cambiare ruolo quando effettui l'accesso come utente IAM, utente del Centro identità IAM, come ruolo federato SAML o come ruolo con federazione delle identità Web.
> - Non è possibile passare da un ruolo AWS Management Console a un ruolo che richiede un ExternalIdvalore. È possibile passare a tale ruolo solo chiamando l'API `AssumeRole` che supporta il parametro `ExternalId`.

- Se l'amministratore ti fornisce un collegamento, scegli il collegamento e passa alla fase Step 5 nella procedura seguente. Il link consente di visualizzare la pagina Web appropriata e di inserire l'ID dell'account (o alias) e il nome del ruolo.
- È possibile costruire manualmente il link e quindi passare alla fase Step 5 nella procedura seguente. Per costruire il collegamento, utilizza il formato seguente:

  `https://signin.aws.amazon.com/switchrole?account=`*`account_id_number`*`&roleName=`*`role_name`*`&displayName=`*`text_to_display`*

  Dove sostituisci il testo seguente:

  - *`account_id_number`*: l'identificazione a 12 cifre dell'account fornito dall'amministratore. In alternativa, l'amministratore può creare un alias dell'account, in modo che l'URL includa il nome dell'account invece dell'ID dell'account. Per ulteriori informazioni, consulta Tipi di utente nella Guida per l'utente di Accedi ad AWS .
  - *`role_name`*: il nome del ruolo che desideri assumere. È possibile ottenerlo dalla fine dell'ARN del ruolo. Ad esempio, fornisce il nome del ruolo `TestRole` dal seguente ruolo ARN: `arn:aws:iam::123456789012:role/TestRole`.
  - (Facoltativo) *`text_to_display`*: il testo che desideri visualizzare nella barra di navigazione al posto del tuo nome utente quando questo ruolo è attivo.
- Puoi cambiare manualmente ruoli utilizzando le informazioni fornite dal tuo amministratore con le procedure seguenti.

Per impostazione predefinita, quando si cambia ruolo, la AWS Management Console sessione dura 1 ora. Le sessioni utente IAM sono 12 ore per impostazione predefinita. Agli utenti IAM viene che cambiano ruoli nella console viene concessa la durata massima della sessione del ruolo o il tempo

rimanente nella sessione dell'utente IAM, a seconda di quale sia minore. Si supponga, ad esempio, che per un ruolo sia impostata una durata massima di sessione di 10 ore. Un utente IAM ha effettuato l'accesso alla console per 8 ore quando decide di cambiare ruolo. Ci sono 4 ore rimanenti nella sessione utente, quindi la durata della sessione ruolo consentita è di 4 ore. Nella tabella seguente viene illustrato come determinare la durata della sessione per un utente IAM quando si cambia ruolo nella console.

Durata della sessione dei ruoli nella console per gli utenti IAM

Il tempo rimanente della sessione utente IAM è...	La durata della sessione del ruolo è…
Meno della durata massima della sessione del ruolo	Tempo rimanente nella sessione utente
Più della durata massima della sessione del ruolo	Valore della durata massima della sessione
Uguale alla durata massima della sessione del ruolo	Valore della durata massima della sessione (approssimativo)

> **Note**
>
> Alcune console AWS di servizio possono rinnovare automaticamente la sessione di ruolo alla scadenza senza che l'utente intraprenda alcuna azione. Alcune potrebbero richiedere di ricaricare la pagina del browser per autenticare nuovamente la sessione.

Per risolvere i problemi più comuni che si possono verificare quando si assume un ruolo, consulta la pagina Non è possibile assumere un ruolo.

Per passare a un ruolo (console)

1. Accedi AWS Management Console come utente IAM e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nella console IAM, scegli il tuo nome utente nella barra di navigazione in alto a destra. Tipicamente appare come questo: ***username (Nome utente)***@***account_ID_number_or_alias***.
3. Seleziona Switch Role (Cambia ruolo). Se è la prima volta che scegli questa opzione, verrà visualizzata una pagina con ulteriori informazioni. Dopo la lettura, scegli Switch Role (Passaggio di ruolo). Se si cancellano i cookie del browser, questa pagina potrebbe apparire di nuovo.
4. Nella pagina Switch Role (Cambia ruolo) inserisci il numero ID dell'account o l'alias dell'account e il nome del ruolo che è stato fornito dall'amministratore.

   > **Note**
   >
   > Se l'amministratore ha creato il ruolo con un percorso, ad esempio `division_abc/subdivision_efg/roleToDoX`, allora è necessario digitare tale percorso completo e il nome nella casella Role (Ruolo). Se digiti solo il nome del ruolo, oppure se il `Path` e il `RoleName` insieme superano 64 caratteri, il passaggio di ruolo fallisce. Si tratta di un limite dei cookie del browser che memorizzano il nome del ruolo. In questo caso, contatta l'amministratore e chiedi di ridurre le dimensioni del percorso e il nome del ruolo.

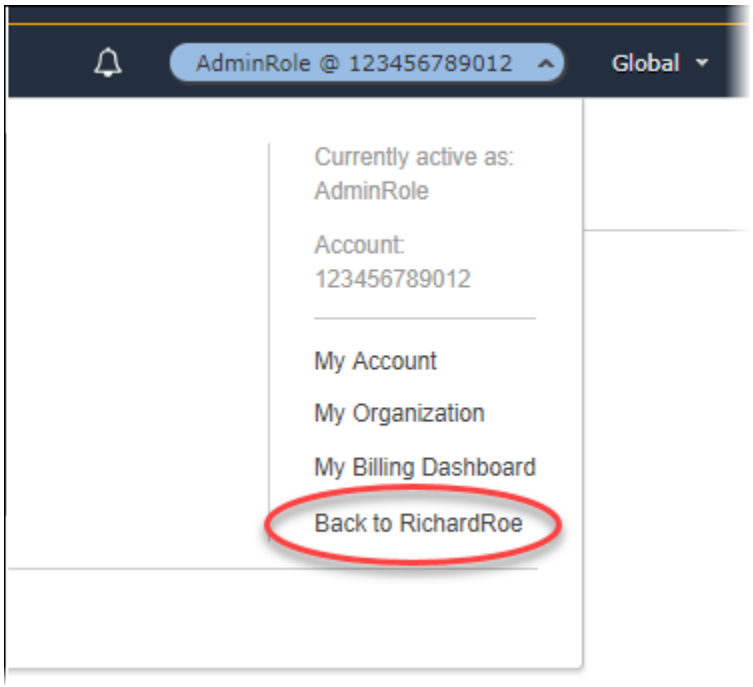
5. (Facoltativo) Scegli un Nome di visualizzazione. (Facoltativo) Inserisci il testo che desideri visualizzare nella barra di navigazione al posto del tuo nome utente quando questo ruolo è attivo. Viene suggerito un nome, in base all'account e alle informazioni del ruolo, ma è possibile modificarlo in base alle proprie esigenze. È inoltre possibile selezionare un colore per evidenziare il nome di visualizzazione. Il nome e il colore possono aiutarti a ricordare quando questo ruolo è attivo, ciò cambia le tue autorizzazioni. Ad esempio, per un ruolo che ti consente di accedere all'ambiente di test, puoi specificare un Nome di visualizzazione uguale a **`Test`** e selezionare il verde in Colore. Per il ruolo che ti consente di accedere all'ambiente di produzione, puoi specificare un Nome di visualizzazione uguale a **`Production`** e selezionare il rosso in Colore.
6. Seleziona Switch Role (Cambia ruolo). Il nome di visualizzazione e il colore sostituiscono il nome utente nella barra di navigazione ed è possibile iniziare a utilizzare le autorizzazioni concesse dal ruolo.

> **Suggerimento**
>
> Gli ultimi ruoli utilizzati appariranno nel menu. La prossima volta che devi passare a uno di questi ruoli, è sufficiente selezionare il ruolo desiderato. Se il ruolo non è visualizzato nel menu, è sufficiente digitare l'account e le informazioni del ruolo manualmente.

Per interrompere l'utilizzo di un ruolo (console)

1. Nella console IAM, scegli il Nome di visualizzazione del tuo ruolo sulla barra di navigazione in alto a destra. Normalmente è simile a questo: ***rolename***@***account_ID_number_or_alias***.
2. Scegli Back to ***username*** (Torna a nome utente). Il ruolo e le relative autorizzazioni sono disattivate e le autorizzazioni associate al tuo utente e ai gruppi IAM vengono automaticamente ripristinate.

   Ad esempio, supponiamo di aver eseguito l'accesso all'account numero `123456789012` utilizzando il nome utente `RichardRoe`. Dopo aver utilizzato il ruolo `AdminRole`, si desidera interrompere l'utilizzo del ruolo e tornare alle autorizzazioni originali. Per smettere di usare un ruolo, scegli AdminRole @ 123456789012, quindi scegli Torna a. RichardRoe



Passaggio a un ruolo IAM (AWS CLI)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse AWS necessarie. In questo senso, è simile a un utente in AWS Identity and Access Management (IAM). Quando effettui l'accesso come utente, ottieni uno specifico set di autorizzazioni. Tuttavia, non effettui l'accesso a un ruolo, ma dopo aver effettuato l'accesso come utente, puoi passare a un ruolo. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Utilizzo di ruoli IAM.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo assunto non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando si assume un ruolo, si lascia temporaneamente l'utente precedente o le autorizzazioni del ruolo e si lavora con le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Puoi utilizzare un ruolo per eseguire un AWS CLI comando quando accedi come utente IAM. Puoi anche utilizzare un ruolo per eseguire un AWS CLI comando quando accedi come utente autenticato esternamente (SAML o OIDC) che sta già utilizzando un ruolo. Inoltre, puoi utilizzare un ruolo per eseguire un comando AWS CLI da un'istanza Amazon EC2 collegata a un ruolo tramite il relativo profilo. Non è possibile assumere un ruolo quando si è effettuato l'accesso come Utente root dell'account AWS.

Concatenamento del ruolo: puoi anche utilizzare la concatenamento dei ruoli che utilizza le autorizzazioni di un ruolo per accedere a un secondo ruolo.

Come impostazione predefinita, la sessione del ruolo dura un'ora. Quando si assume questo ruolo utilizzando le operazioni della CLI `assume-role*`, è possibile specificare un valore per il parametro `duration-seconds`. Questo valore può variare da 900 secondi (15 minuti) fino alla durata massima della sessione per il ruolo. Se cambi ruolo nella console, la durata della sessione è limitata a un massimo di un'ora. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.

Se si utilizza il concatenamento dei ruoli, la tua durata della sessione è limitata a un massimo di un'ora. Se successivamente utilizzi il parametro `duration-seconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

Scenario di esempio: passaggio a un ruolo di produzione

Immagina di essere un utente IAM per utilizzare l'ambiente di sviluppo. In questo scenario, a volte è necessario utilizzare l'ambiente di produzione nella riga di comando con l'AWS CLI. Disponi già di un set di credenziali con chiave di accesso. Questa può essere la coppia di chiavi di accesso assegnata all'utente IAM standard. Oppure, se hai effettuato l'accesso come un utente federato, può essere la coppia di chiavi di accesso per il ruolo che ti è stato inizialmente assegnato. Se le autorizzazioni attuali ti garantiscono la possibilità di assumere un ruolo IAM specifico, puoi identificare quel ruolo in un «profilo» nei file di configurazione. AWS CLI Questo comando viene quindi eseguito con le autorizzazioni del ruolo IAM specificato, non con l'identità originale. Nota che quando specifichi quel profilo in un AWS CLI comando, stai usando il nuovo ruolo. In questa situazione, non puoi utilizzare le autorizzazioni originali nell'account di sviluppo nello stesso momento. Il motivo è che solo un set di autorizzazioni può essere attivo alla volta.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i registri per scoprire chi ha eseguito un'azione in. AWS L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consulta `sts:SourceIdentity` e `sts:RoleSessionName`.

Per passare a un ruolo di produzione (AWS CLI)

1. Se non hai mai usato il AWS CLI, devi prima configurare il tuo profilo CLI predefinito. Apri un prompt dei comandi e configura AWS CLI l'installazione per utilizzare la chiave di accesso del tuo utente IAM o del tuo ruolo federato. Per ulteriori informazioni, consulta Configurazione della AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface .

   Esegui il comando aws configure come riportato di seguito:

   ```
   aws configure
   ```

   Quando viene richiesto, fornire le seguenti informazioni:

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-east-2
Default output format [None]: json
```

2. Creare un nuovo profilo per il ruolo nel file `.aws/config` in Unix o Linux, oppure nel file `C:\Users\USERNAME\.aws\config` in Windows. L'esempio seguente crea un profilo denominato `prodaccess` che passa al ruolo *`ProductionAccessRole`* nell'account `123456789012`. L'ARN del ruolo si ottiene dall'amministratore dell'account che ha creato il ruolo. Quando questo profilo viene richiamato, AWS CLI utilizza le credenziali di `source_profile` per richiedere le credenziali per il ruolo. Per questo motivo, l'identità alla quale viene fatto riferimento come `source_profile` deve disporre delle autorizzazioni `sts:AssumeRole` per il ruolo specificato in `role_arn`.

   ```
   [profile prodaccess]
       role_arn = arn:aws:iam::123456789012:role/ProductionAccessRole
       source_profile = default
   ```

3. Dopo aver creato il nuovo profilo, qualsiasi AWS CLI comando che specifica il parametro `--profile prodaccess` viene eseguito con le autorizzazioni associate al ruolo IAM `ProductionAccessRole` anziché con l'utente predefinito.

   ```
   aws iam list-users --profile prodaccess
   ```

   Questo comando funziona se le autorizzazioni assegnate a `ProductionAccessRole` permettono di elencare gli utenti nell'account attuale AWS .

4. Per ripristinare le autorizzazioni concesse dalle credenziali originali, eseguire i comandi senza il parametro `--profile`. Torna a AWS CLI utilizzare le credenziali nel tuo profilo predefinito, in cui hai configurato. Step 1

Per ulteriori informazioni, consulta Assunzione di un ruolo nella Guida per l'utente di AWS Command Line Interface .

Scenario di esempio: consentire a un ruolo del profilo dell'istanza di passare a un ruolo in un altro account

Immagina di usarne due Account AWS e di voler consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire AWS CLIcomandi in entrambi gli account. Supponiamo che

l'istanza EC2 esista nell'account 111111111111. Tale istanza include il ruolo del profilo dell'istanza abcd che consente all'applicazione di eseguire attività Amazon S3 di sola lettura nel bucket my-bucket-1 all'interno dello stesso account 111111111111. Tuttavia, l'applicazione deve anche poter assumere il ruolo tra più account efgh per eseguire attività nell'account 222222222222. A questo scopo, il ruolo del profilo dell'istanza EC2 abcd deve disporre della policy di autorizzazioni seguente:

Policy di autorizzazioni del ruolo ***abcd*** 111111111111 dell'account

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket-1/*",
                "arn:aws:s3:::my-bucket-1"
            ]
        },
        {
            "Sid": "AllowIPToAssumeCrossAccountRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::222222222222:role/efgh"
        }
    ]
}
```

Supponiamo che il ruolo tra account `efgh` consenta attività Amazon S3 di sola lettura nel bucket `my-bucket-2` all'interno dello stesso account `222222222222`. A tale scopo, il ruolo tra account `efgh` deve disporre della seguente policy di autorizzazioni:

Policy di autorizzazioni del ruolo ***efgh*** 222222222222 dell'account

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket-2/*",
                "arn:aws:s3:::my-bucket-2"
            ]
        }
    ]
}
```

Il ruolo `efgh` deve consentire al ruolo del profilo dell'istanza `abcd` di assumerlo. A tale scopo, il ruolo `efgh` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***efgh*** dell'account 222222222222

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "efghTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111111111111:role/abcd"}
        }
    ]
}
```

Per eseguire quindi AWS CLI i comandi nell'account222222222222, è necessario aggiornare il file di configurazione CLI. Identifica il ruolo `efgh` come il "profilo" e il ruolo del profilo dell'istanza EC2 `abcd` come "origine delle credenziali" nel file di configurazione di AWS CLI . I comandi della CLI vengono quindi eseguiti con le autorizzazioni del ruolo `efgh`, non il ruolo `abcd` originale.

> **Note**
>
> Per motivi di sicurezza, è possibile utilizzare AWS CloudTrail per controllare l'uso dei ruoli nell'account. Per distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da diversi responsabili nei CloudTrail log, è possibile utilizzare il nome della sessione di ruolo. Quando AWS CLI assume un ruolo per conto di un utente come descritto in questo argomento, viene creato automaticamente un nome di sessione di ruolo come. `AWS-CLI-session-`*`nnnnnnnn`* Di seguito *`nnnnnnnn`* è un intero che rappresenta il tempo in Tempo Unix epoch (il numero di secondi dalla mezzanotte UTC il 1° gennaio 1970). Per ulteriori informazioni, consulta CloudTrail Event Reference nella Guida per l'AWS CloudTrail utente.

Per consentire a un ruolo del profilo dell'istanza EC2 di passare a un ruolo tra account (AWS CLI)

1. Non è necessario configurare un profilo predefinito della CLI. Al contrario, puoi caricare le credenziali dai metadati del profilo dell'istanza EC2. Crea un nuovo profilo per il ruolo nel file `.aws/config`. L'esempio seguente crea un profilo `instancecrossaccount` che passa al ruolo *`efgh`* nell'account 222222222222. Quando questo profilo viene richiamato, AWS CLI utilizza le credenziali dei metadati del profilo dell'istanza EC2 per richiedere le credenziali per il ruolo. Per questo motivo, il ruolo del profilo dell'istanza EC2 deve disporre delle autorizzazioni `sts:AssumeRole` per il ruolo specificato nel `role_arn`.

   ```
   [profile instancecrossaccount]
   role_arn = arn:aws:iam::222222222222:role/efgh
   credential_source = Ec2InstanceMetadata
   ```

2. Dopo aver creato il nuovo profilo, qualsiasi AWS CLI comando che specifica il parametro `--profile instancecrossaccount` viene eseguito con le autorizzazioni associate al `efgh` ruolo nell'account. `222222222222`

   ```
   aws s3 ls my-bucket-2 --profile instancecrossaccount
   ```

   Questo comando funziona se le autorizzazioni che vengono assegnate al ruolo `efgh` consentono di elencare gli utenti nell' Account AWS corrente.
3. Per tornare alle autorizzazioni del profilo dell'istanza EC2 originale nell'account `111111111111`, esegui i comandi della CLI senza il parametro `--profile`.

Per ulteriori informazioni, consulta Assunzione di un ruolo nella Guida per l'utente di AWS Command Line Interface .

Passaggio a un ruolo IAM (Tools for Windows PowerShell)

Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse AWS necessarie. In questo senso, è simile a un utente in AWS Identity and Access Management (IAM). Quando effettui l'accesso come utente, ottieni uno specifico set di autorizzazioni. Tuttavia, non accedi a un ruolo, ma una volta effettuato l'accesso puoi passare a un ruolo. Ciò consente di accantonare temporaneamente le autorizzazioni utente originali e usufruire invece delle autorizzazioni assegnate al ruolo. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo a cui si passa non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando passi a un ruolo, lasci temporaneamente le autorizzazioni utente e utilizzi le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni utente vengono automaticamente ripristinate.

Questa sezione descrive come cambiare ruoli quando utilizzi la riga di comando con gli AWS Tools for Windows PowerShell.

Immagina di avere un account nell'ambiente di sviluppo e di dover occasionalmente lavorare con l'ambiente di produzione dalla riga di comando utilizzando gli strumenti per Windows PowerShell.

Disponi già di un set di credenziali con chiave di accesso. Può trattarsi di una coppia di chiavi di accesso assegnata all'utente IAM standard. In alternativa, se hai effettuato l'accesso come utente federato, può trattarsi della coppia di chiavi di accesso per il ruolo inizialmente assegnato. Puoi utilizzare queste credenziali per eseguire il cmdlet `Use-STSRole` che passa l'ARN di un nuovo ruolo come parametro. Il comando restituisce le credenziali di sicurezza temporanee per il ruolo richiesto. È quindi possibile utilizzare tali credenziali nei PowerShell comandi successivi con le autorizzazioni del ruolo per accedere alle risorse in produzione. Mentre utilizzi il ruolo, non puoi utilizzare le autorizzazioni utente dell'account di sviluppo perché è attivo un solo set di autorizzazioni alla volta.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i registri per scoprire chi ha eseguito un'azione in. AWS L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consulta `sts:SourceIdentity` e `sts:RoleSessionName`.

Tutte le chiavi di accesso e i token sono solo esempi e non possono essere utilizzati come mostrato. Sostituiscili con i valori appropriati del tuo ambiente reale.

Per passare a un ruolo (Strumenti per Windows) PowerShell

1. Apri un PowerShell prompt dei comandi e configura il profilo predefinito per utilizzare la chiave di accesso del tuo attuale utente IAM o del tuo ruolo federato. Se in precedenza hai utilizzato gli Strumenti per Windows PowerShell, probabilmente l'operazione è già stata eseguita. Tieni presente che è possibile cambiare ruolo solo se hai effettuato l'accesso come utente IAM e non come Utente root dell'account AWS.

   ```
   PS C:\> Set-AWSCredentials -AccessKey AKIAIOSFODNN7EXAMPLE -
   SecretKey wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY -StoreAs MyMainUserProfile
   PS C:\> Initialize-AWSDefaults -ProfileName MyMainUserProfile -Region us-east-2
   ```

   Per ulteriori informazioni, vedere Utilizzo AWS delle credenziali nella Guida per l'AWS Tools for Windows PowerShell utente.
2. Per recuperare le credenziali per il nuovo ruolo, eseguire il comando seguente per passare al ruolo *RoleName* nell'account 123456789012. L'ARN del ruolo si ottiene dall'amministratore

dell'account che ha creato il ruolo. Il comando richiede di fornire anche un nome di sessione. È possibile selezionare qualsiasi testo. Il comando seguente richiede le credenziali e quindi acquisisce l'oggetto proprietà `Credentials` dall'oggetto risultati restituiti e lo memorizza nella variabile `$Creds`.

```
PS C:\> $Creds = (Use-STSRole -RoleArn "arn:aws:iam::123456789012:role/RoleName" -
RoleSessionName "MyRoleSessionName").Credentials
```

`$Creds` è un oggetto che ora contiene gli elementi `AccessKeyId`, `SecretAccessKey` e `SessionToken` necessari nelle fasi successive. I seguenti comandi di esempio illustrano valori tipici:

```
PS C:\> $Creds.AccessKeyId
AKIAIOSFODNN7EXAMPLE

PS C:\> $Creds.SecretAccessKey
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

PS C:\> $Creds.SessionToken
AQoDYXdzEGcaEXAMPLE2gsYULo
+Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLECvSRyh0FW7jEXAMPLEW+vE/7s1HRp
XviG7b+qYf4nD00EXAMPLEmj4wxS04L/uZEXAMPLECihzFB5lTYLto9dyBgSDyEXAMPLE9/
g7QRUhZp4bqbEXAMPLENwGPy
Oj59pFA4lNKCIkVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3UuysgsKdEXAMPLE1TVastU1A0SKFEXAMPLEiyw
C
s8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP+4eZScEXAMPLEsnf87eNhyDHq6ikBQ==

PS C:\> $Creds.Expiration
Thursday, June 18, 2018 2:28:31 PM
```

3. Per utilizzare queste credenziali per ogni successivo comando, includerle con il parametro `-Credential`. Ad esempio, il comando seguente utilizza le credenziali del ruolo e funziona solo se al ruolo viene concessa l'autorizzazione `iam:ListRoles` grazie alla quale può quindi eseguire il cmdlet `Get-IAMRoles`:

```
PS C:\> get-iamroles -Credential $Creds
```

4. Per tornare alle credenziali originali, è sufficiente interrompere l'utilizzo del `-Credentials $Creds` parametro e consentire PowerShell il ripristino delle credenziali archiviate nel profilo predefinito.

Passaggio a un ruolo IAM (AWS API)

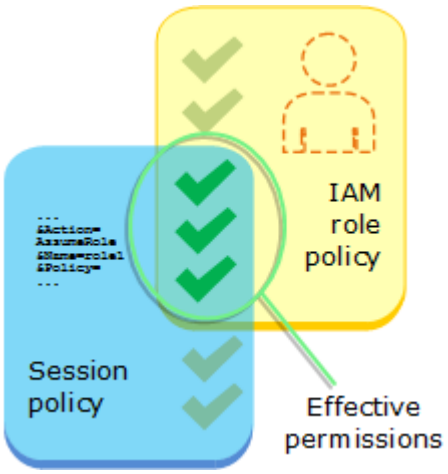
Un ruolo specifica un set di autorizzazioni da utilizzare per accedere alle risorse di AWS . In questo senso, è simile a un utente IAM. Un principale (persona o applicazione) assume il ruolo di ricevere le autorizzazioni temporanee per svolgere le attività richieste e interagire con AWS le risorse. Il ruolo può trovarsi nel tuo account o in qualsiasi altro Account AWS. Per ulteriori informazioni sui ruoli e i relativi vantaggi e su come crearli e configurarli, consulta Ruoli IAM e Creazione di ruoli IAM. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Utilizzo di ruoli IAM.

> **Important**
>
> Le autorizzazioni dell'utente IAM e di qualsiasi ruolo assunto non sono cumulative. Un solo set di autorizzazioni è attivo alla volta. Quando si assume un ruolo, si lascia temporaneamente l'utente precedente o le autorizzazioni del ruolo e si lavora con le autorizzazioni assegnate al ruolo. Quando lasci il ruolo, le autorizzazioni originali vengono automaticamente ripristinate.

Per assumere un ruolo, un'applicazione chiama l'operazione AWS STS `AssumeRole`API e passa l'ARN del ruolo da utilizzare. L'operazione crea una nuova sessione con le credenziali temporanee. Questa sessione ha le stesse autorizzazioni delle policy basate su identità per quel ruolo.

Quando chiami `AssumeRole`, puoi passare facoltativamente policy di sessione inline o gestite. Le policy di sessione sono policy avanzate che vengono passate come un parametro quando si crea una sessione temporanea a livello di programma per un ruolo o un utente federato. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione delle policy basate sull'identità dell'entità e delle policy di sessione. Le policy di sessione sono utili quando occorre fornire le credenziali temporanee del ruolo a un'altra persona, che potrà usare le credenziali temporanee del ruolo nelle chiamate API AWS successive, per accedere alle risorse nell'account che possiede il ruolo. Non è possibile utilizzare policy di sessione per concedere autorizzazioni maggiori rispetto a quelle

consentite dalla policy basata su identità. Per ulteriori informazioni su come AWS determina le autorizzazioni effettive di un ruolo, consulta. Logica di valutazione delle policy



Per chiamare `AssumeRole` devi aver effettuato l'accesso come utente IAM oppure come utente autenticato esternamente (SAML oppure OIDC) e utilizzare già un ruolo. Puoi anche utilizzare una concatenazione dei ruoli ovvero usare un ruolo per definirne un secondo. Non è possibile assumere un ruolo quando si è effettuato l'accesso come Utente root dell'account AWS.

Come impostazione predefinita, la sessione del ruolo dura un'ora. Quando assumi questo ruolo utilizzando le operazioni AWS STS `AssumeRole*`API, puoi specificare un valore per il `DurationSeconds` parametro. Questo valore può variare da 900 secondi (15 minuti) fino alla durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.

Se scegli di ricorrere alla concatenazione dei ruoli, la durata della sessione è limitata a un'ora. Se successivamente utilizzi il parametro `DurationSeconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

> **Note**
>
> Per motivi di sicurezza, gli amministratori possono esaminare AWS CloudTrail i log per scoprire chi ha eseguito un'azione in. AWS L'amministratore potrebbe richiedere di specificare una identità di origine o un nome della sessione del ruolo quando si assume il ruolo. Per ulteriori informazioni, consulta `sts:SourceIdentity` e `sts:RoleSessionName`.

Gli esempi di codice seguenti mostrano come creare un utente e assumere un ruolo.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Crea un utente che non disponga di autorizzazioni.
- Crea un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3 per l'account.
- Aggiungi una policy per consentire all'utente di assumere il ruolo.
- Assumi il ruolo ed elenca i bucket S3 utilizzando le credenziali temporanee, quindi ripulisci le risorse.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
global using Amazon.IdentityManagement;
global using Amazon.S3;
global using Amazon.SecurityToken;
global using IAMActions;
global using IamScenariosCommon;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;


namespace IAMActions;

public class IAMWrapper
{
```

```
    private readonly IAmazonIdentityManagementService _IAMService;

    /// <summary>
    /// Constructor for the IAMWrapper class.
    /// </summary>
    /// <param name="IAMService">An IAM client object.</param>
    public IAMWrapper(IAmazonIdentityManagementService IAMService)
    {
        _IAMService = IAMService;
    }

    /// <summary>
    /// Add an existing IAM user to an existing IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to add.</param>
    /// <param name="groupName">The name of the group to add the user to.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AddUserToGroupAsync(string userName, string
 groupName)
    {
        var response = await _IAMService.AddUserToGroupAsync(new
 AddUserToGroupRequest
        {
            GroupName = groupName,
            UserName = userName,
        });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
```

```
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }


    /// <summary>
    /// Create an IAM group.
    /// </summary>
    /// <param name="groupName">The name to give the IAM group.</param>
    /// <returns>The IAM group that was created.</returns>
    public async Task<Group> CreateGroupAsync(string groupName)
    {
        var response = await _IAMService.CreateGroupAsync(new CreateGroupRequest
{ GroupName = groupName });
        return response.Group;
    }


    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
```

```
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });

        return response.Policy;
    }


    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
 rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }


    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
```

```
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }


    /// <summary>
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });
```

```
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupAsync(string groupName)
    {
        var response = await _IAMService.DeleteGroupAsync(new DeleteGroupRequest
{ GroupName = groupName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM policy associated with an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group associated with the
    /// policy.</param>
    /// <param name="policyName">The name of the policy to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupPolicyAsync(string groupName, string
policyName)
    {
        var request = new DeleteGroupPolicyRequest()
        {
            GroupName = groupName,
            PolicyName = policyName,
        };

        var response = await _IAMService.DeleteGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM policy.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
```

```
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
 DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
```

```
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
{ UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user policy.
    /// </summary>
    /// <param name="policyName">The name of the IAM policy to delete.</param>
    /// <param name="userName">The username of the IAM user.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Detach an IAM policy from an IAM role.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
```

```
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
 GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }


    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }


    /// <summary>
    /// Get information about an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to retrieve information
    /// for.</param>
    /// <returns>The IAM role that was retrieved.</returns>
    public async Task<Role> GetRoleAsync(string roleName)
    {
        var response = await _IAMService.GetRoleAsync(new GetRoleRequest
        {
            RoleName = roleName,
```

```
        });

        return response.Role;
    }


    /// <summary>
    /// Get information about an IAM user.
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }


    /// <summary>
    /// List IAM groups.
    /// </summary>
```

```
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
 ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }


    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }


    /// <summary>
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
```

```
        var listRolePoliciesPaginator =
_IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }


    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }


    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }
```

```
    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Remove a user from an IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to remove.</param>
    /// <param name="groupName">The name of the IAM group to remove the user
from.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> RemoveUserFromGroupAsync(string userName, string
groupName)
    {
        // Remove the user from the group.
        var removeUserRequest = new RemoveUserFromGroupRequest()
        {
            UserName = userName,
            GroupName = groupName,
        };

        var response = await
_IAMService.RemoveUserFromGroupAsync(removeUserRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
```

```
    /// Add or update an inline policy document that is embedded in an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
 policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutGroupPolicyAsync(string groupName, string
 policyName, string policyDocument)
    {
        var request = new PutGroupPolicyRequest
        {
            GroupName = groupName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

```
   /// <summary>
   /// Add or update an inline policy document that is embedded in an IAM user.
   /// </summary>
   /// <param name="userName">The name of the IAM user.</param>
   /// <param name="policyName">The name of the IAM policy.</param>
   /// <param name="policyDocument">The policy document defining the IAM
policy.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> PutUserPolicyAsync(string userName, string
policyName, string policyDocument)
   {
       var request = new PutUserPolicyRequest
       {
           UserName = userName,
           PolicyName = policyName,
           PolicyDocument = policyDocument
       };

       var response = await _IAMService.PutUserPolicyAsync(request);
       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }

   /// <summary>
   /// Wait for a new access key to be ready to use.
   /// </summary>
   /// <param name="accessKeyId">The Id of the access key.</param>
   /// <returns>A boolean value indicating the success of the action.</returns>
   public async Task<bool> WaitUntilAccessKeyIsReady(string accessKeyId)
   {
       var keyReady = false;

       do
       {
           try
           {
               var response = await _IAMService.GetAccessKeyLastUsedAsync(
                   new GetAccessKeyLastUsedRequest { AccessKeyId =
accessKeyId });
               if (response.UserName is not null)
               {
                   keyReady = true;
               }
           }
```

```
            catch (NoSuchEntityException)
            {
                keyReady = false;
            }
        } while (!keyReady);

        return keyReady;
    }
}



using Microsoft.Extensions.Configuration;

namespace IAMBasics;

public class IAMBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the AWS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
            .AddTransient<IAMWrapper>()
            .AddTransient<UIWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<IAMBasics>();


        IConfiguration configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
```

```
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // Values needed for user, role, and policies.
        string userName = configuration["UserName"]!;
        string s3PolicyName = configuration["S3PolicyName"]!;
        string roleName = configuration["RoleName"]!;


        var iamWrapper = host.Services.GetRequiredService<IAMWrapper>();
        var uiWrapper = host.Services.GetRequiredService<UIWrapper>();

        uiWrapper.DisplayBasicsOverview();
        uiWrapper.PressEnter();

        // First create a user. By default, the new user has
        // no permissions.
        uiWrapper.DisplayTitle("Create User");
        Console.WriteLine($"Creating a new user with user name: {userName}.");
        var user = await iamWrapper.CreateUserAsync(userName);
        var userArn = user.Arn;

        Console.WriteLine($"Successfully created user: {userName} with ARN:
{userArn}.");
        uiWrapper.WaitABit(15, "Now let's wait for the user to be ready for
use.");

        // Define a role policy document that allows the new user
        // to assume the role.
        string assumeRolePolicyDocument = "{" +
          "\"Version\": \"2012-10-17\"," +
          "\"Statement\": [{" +
              "\"Effect\": \"Allow\"," +
              "\"Principal\": {" +
              $" \"AWS\": \"{userArn}\"" +
              "}," +
              "\"Action\": \"sts:AssumeRole\"" +
          "}]" +
        "}";

        // Permissions to list all buckets.
        string policyDocument = "{" +
            "\"Version\": \"2012-10-17\"," +
```

```
            " \"Statement\" : [{" +
                " \"Action\" : [\"s3:ListAllMyBuckets\"]," +
                " \"Effect\" : \"Allow\"," +
                " \"Resource\" : \"*\"" +
            "}]" +
        "}";

        // Create an AccessKey for the user.
        uiWrapper.DisplayTitle("Create access key");
        Console.WriteLine("Now let's create an access key for the new user.");
        var accessKey = await iamWrapper.CreateAccessKeyAsync(userName);

        var accessKeyId = accessKey.AccessKeyId;
        var secretAccessKey = accessKey.SecretAccessKey;

        Console.WriteLine($"We have created the access key with Access key id:
{accessKeyId}.");

        Console.WriteLine("Now let's wait until the IAM access key is ready to
use.");
        var keyReady = await iamWrapper.WaitUntilAccessKeyIsReady(accessKeyId);

        // Now try listing the Amazon Simple Storage Service (Amazon S3)
        // buckets. This should fail at this point because the user doesn't
        // have permissions to perform this task.
        uiWrapper.DisplayTitle("Try to display Amazon S3 buckets");
        Console.WriteLine("Now let's try to display a list of the user's Amazon
S3 buckets.");
        var s3Client1 = new AmazonS3Client(accessKeyId, secretAccessKey);
        var stsClient1 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        var s3Wrapper = new S3Wrapper(s3Client1, stsClient1);
        var buckets = await s3Wrapper.ListMyBucketsAsync();

        Console.WriteLine(buckets is null
            ? "As expected, the call to list the buckets has returned a null
list."
            : "Something went wrong. This shouldn't have worked.");

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Create IAM role");
        Console.WriteLine($"Creating the role: {roleName}");
```

```
        // Creating an IAM role to allow listing the S3 buckets. A role name
        // is not case sensitive and must be unique to the account for which it
        // is created.
        var roleArn = await iamWrapper.CreateRoleAsync(roleName,
assumeRolePolicyDocument);

        uiWrapper.PressEnter();

        // Create a policy with permissions to list S3 buckets.
        uiWrapper.DisplayTitle("Create IAM policy");
        Console.WriteLine($"Creating the policy: {s3PolicyName}");
        Console.WriteLine("with permissions to list the Amazon S3 buckets for the
account.");
        var policy = await iamWrapper.CreatePolicyAsync(s3PolicyName,
policyDocument);

        // Wait 15 seconds for the IAM policy to be available.
        uiWrapper.WaitABit(15, "Waiting for the policy to be available.");

        // Attach the policy to the role you created earlier.
        uiWrapper.DisplayTitle("Attach new IAM policy");
        Console.WriteLine("Now let's attach the policy to the role.");
        await iamWrapper.AttachRolePolicyAsync(policy.Arn, roleName);

        // Wait 15 seconds for the role to be updated.
        Console.WriteLine();
        uiWrapper.WaitABit(15, "Waiting for the policy to be attached.");

        // Use the AWS Security Token Service (AWS STS) to have the user
        // assume the role we created.
        var stsClient2 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        // Wait for the new credentials to become valid.
        uiWrapper.WaitABit(10, "Waiting for the credentials to be valid.");

        var assumedRoleCredentials = await
s3Wrapper.AssumeS3RoleAsync("temporary-session", roleArn);

        // Try again to list the buckets using the client created with
        // the new user's credentials. This time, it should work.
        var s3Client2 = new AmazonS3Client(assumedRoleCredentials);
```

```
        s3Wrapper.UpdateClients(s3Client2, stsClient2);

        buckets = await s3Wrapper.ListMyBucketsAsync();

        uiWrapper.DisplayTitle("List Amazon S3 buckets");
        Console.WriteLine("This time we should have buckets to list.");
        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
            {
                Console.WriteLine($"{bucket.BucketName} created:
 {bucket.CreationDate}");
            });
        }

        uiWrapper.PressEnter();

        // Now clean up all the resources used in the example.
        uiWrapper.DisplayTitle("Clean up resources");
        Console.WriteLine("Thank you for watching. The IAM Basics demo is
 complete.");
        Console.WriteLine("Please wait while we clean up the resources we
 created.");

        await iamWrapper.DetachRolePolicyAsync(policy.Arn, roleName);

        await iamWrapper.DeletePolicyAsync(policy.Arn);

        await iamWrapper.DeleteRoleAsync(roleName);

        await iamWrapper.DeleteAccessKeyAsync(accessKeyId, userName);

        await iamWrapper.DeleteUserAsync(userName);

        uiWrapper.PressEnter();

        Console.WriteLine("All done cleaning up our resources. Thank you for your
 patience.");
    }
}


namespace IamScenariosCommon;
```

```
using System.Net;

/// <summary>
/// A class to perform Amazon Simple Storage Service (Amazon S3) actions for
/// the IAM Basics scenario.
/// </summary>
public class S3Wrapper
{
    private IAmazonS3 _s3Service;
    private IAmazonSecurityTokenService _stsService;

    /// <summary>
    /// Constructor for the S3Wrapper class.
    /// </summary>
    /// <param name="s3Service">An Amazon S3 client object.</param>
    /// <param name="stsService">An AWS Security Token Service (AWS STS)
    /// client object.</param>
    public S3Wrapper(IAmazonS3 s3Service, IAmazonSecurityTokenService stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }

    /// <summary>
    /// Assumes an AWS Identity and Access Management (IAM) role that allows
    /// Amazon S3 access for the current session.
    /// </summary>
    /// <param name="roleSession">A string representing the current session.</
param>
    /// <param name="roleToAssume">The name of the IAM role to assume.</param>
    /// <returns>Credentials for the newly assumed IAM role.</returns>
    public async Task<Credentials> AssumeS3RoleAsync(string roleSession, string
 roleToAssume)
    {
        // Create the request to use with the AssumeRoleAsync call.
        var request = new AssumeRoleRequest()
        {
            RoleSessionName = roleSession,
            RoleArn = roleToAssume,
        };

        var response = await _stsService.AssumeRoleAsync(request);

        return response.Credentials;
```

```
    }


    /// <summary>
    /// Delete an S3 bucket.
    /// </summary>
    /// <param name="bucketName">Name of the S3 bucket to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteBucketAsync(string bucketName)
    {
        var result = await _s3Service.DeleteBucketAsync(new DeleteBucketRequest
{ BucketName = bucketName });
        return result.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the buckets that are owned by the user's account.
    /// </summary>
    /// <returns>Async Task.</returns>
    public async Task<List<S3Bucket>?> ListMyBucketsAsync()
    {
        try
        {
            // Get the list of buckets accessible by the new user.
            var response = await _s3Service.ListBucketsAsync();

            return response.Buckets;
        }
        catch (AmazonS3Exception ex)
        {
            // Something else went wrong. Display the error message.
            Console.WriteLine($"Error: {ex.Message}");
            return null;
        }
    }

    /// <summary>
    /// Create a new S3 bucket.
    /// </summary>
    /// <param name="bucketName">The name for the new bucket.</param>
    /// <returns>A Boolean value indicating whether the action completed
    /// successfully.</returns>
    public async Task<bool> PutBucketAsync(string bucketName)
    {
```

```
        var response = await _s3Service.PutBucketAsync(new PutBucketRequest
 { BucketName = bucketName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Update the client objects with new client objects. This is available
    /// because the scenario uses the methods of this class without and then
    /// with the proper permissions to list S3 buckets.
    /// </summary>
    /// <param name="s3Service">The Amazon S3 client object.</param>
    /// <param name="stsService">The AWS STS client object.</param>
    public void UpdateClients(IAmazonS3 s3Service, IAmazonSecurityTokenService
 stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }
}


namespace IamScenariosCommon;

public class UIWrapper
{
    public readonly string SepBar = new('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the IAM Groups scenario.
    /// </summary>
    public void DisplayGroupsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to the IAM Groups Demo");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates an Amazon Identity and Access Management
 (IAM) group.");
        Console.WriteLine("\t2. Adds an IAM policy to the IAM group giving it
 full access to Amazon S3.");
        Console.WriteLine("\t3. Creates a new IAM user.");
        Console.WriteLine("\t4. Creates an IAM access key for the user.");
        Console.WriteLine("\t5. Adds the user to the IAM group.");
        Console.WriteLine("\t6. Lists the buckets on the account.");
```

```
        Console.WriteLine("\t7. Proves that the user has full Amazon S3 access by
creating a bucket.");
        Console.WriteLine("\t8. List the buckets again to show the new bucket.");
        Console.WriteLine("\t9. Cleans up all the resources created.");
    }

    /// <summary>
    /// Show information about the IAM Basics scenario.
    /// </summary>
    public void DisplayBasicsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to IAM Basics");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates a user with no permissions.");
        Console.WriteLine("\t2. Creates a role and policy that grant
s3:ListAllMyBuckets permission.");
        Console.WriteLine("\t3. Grants the user permission to assume the role.");
        Console.WriteLine("\t4. Creates an S3 client object as the user and tries
to list buckets (this will fail).");
        Console.WriteLine("\t5. Gets temporary credentials by assuming the
role.");
        Console.WriteLine("\t6. Creates a new S3 client object with the temporary
credentials and lists the buckets (this will succeed).");
        Console.WriteLine("\t7. Deletes all the resources.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPress <Enter> to continue. ");
        _ = Console.ReadLine();
        Console.WriteLine();
    }

    /// <summary>
    /// Pad a string with spaces to center it on the console display.
    /// </summary>
    /// <param name="strToCenter">The string to be centered.</param>
    /// <returns>The padded string.</returns>
    public string CenterString(string strToCenter)
```

```
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title, and another
    /// line of hyphens.
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }

    /// <summary>
    /// Display a countdown and wait for a number of seconds.
    /// </summary>
    /// <param name="numSeconds">The number of seconds to wait.</param>
    public void WaitABit(int numSeconds, string msg)
    {
        Console.WriteLine(msg);

        // Wait for the requested number of seconds.
        for (int i = numSeconds; i > 0; i--)
        {
            System.Threading.Thread.Sleep(1000);
            Console.Write($"{i}...");
        }

        PressEnter();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - AttachRolePolitica

- CreateAccessChiave
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessChiave
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might
 be necessary to
#           create a custom policy).
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
```

```
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role
 to the user."
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result

  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}
```

```
 echo "Created access key named $key_name"

 echo "Wait 10 seconds for the user to be ready."
 sleep 10
 echo_repeat "*" 88
 echo

 local iam_role_name
 iam_role_name=$(generate_random_name "test-role")
 echo "Creating a role named $iam_role_name with user $user_name as the
principal."

 local assume_role_policy_document="{
   \"Version\": \"2012-10-17\",
   \"Statement\": [{
       \"Effect\": \"Allow\",
       \"Principal\": {\"AWS\": \"$user_arn\"},
       \"Action\": \"sts:AssumeRole\"
       }]
   }"

 local role_arn
 role_arn=$(iam_create_role -n "$iam_role_name" -p
"$assume_role_policy_document")

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Created IAM role named $iam_role_name"
 else
   errecho "The role failed to create. This demo will exit."
   clean_up "$user_name" "$key_name"
   return 1
 fi

 local policy_name
 policy_name=$(generate_random_name "test-policy")
 local policy_document="{
              \"Version\": \"2012-10-17\",
              \"Statement\": [{
                  \"Effect\": \"Allow\",
                  \"Action\": \"s3:ListAllMyBuckets\",
                  \"Resource\": \"arn:aws:s3:::*\"}]}"
```

```
  local policy_arn
  policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created  IAM policy named $policy_name"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
  fi

  if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
  else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
  fi

  local assume_role_policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"$role_arn\"}]}"

  local assume_role_policy_name
  assume_role_policy_name=$(generate_random_name "test-assume-role-")

  # shellcheck disable=SC2181
  local assume_role_policy_arn
  assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn"
    return 1
  fi
```

```
 echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
 sleep 10
 echo_repeat "*" 88
 echo

 echo "Try to list buckets without the new user assuming the role."
 echo_repeat "*" 88
 echo

 # Set the environment variables for the created user.
 # bashsupport disable=BP2001
 export AWS_ACCESS_KEY_ID=$key_name
 # bashsupport disable=BP2001
 export AWS_SECRET_ACCESS_KEY=$key_secret

 local buckets
 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
   echo "There are $bucket_count buckets in the account. This should not have
happened."
 else
   errecho "Because the role with permissions has not been assumed, listing
buckets failed."
 fi

 echo
 echo_repeat "*" 88
 echo "Now assume the role $iam_role_name and list the buckets."
 echo_repeat "*" 88
 echo

 local credentials

 credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Assumed role $iam_role_name"
 else
   errecho "Failed to assume role."
```

```
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
    return 1
  fi

  IFS=$'\t ' read -r -a credentials <<<"$credentials"

  export AWS_ACCESS_KEY_ID=${credentials[0]}
  export AWS_SECRET_ACCESS_KEY=${credentials[1]}
  # bashsupport disable=BP2001
  export AWS_SESSION_TOKEN=${credentials[2]}

  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets
succeeded because of "
    echo "the assumed role."
  else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
    return 1
  fi

  local result=0
  export AWS_ACCESS_KEY_ID=""
  export AWS_SECRET_ACCESS_KEY=""

  echo
  echo_repeat "*" 88
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo
```

```
  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

Le funzioni IAM utilizzate in questo scenario.

```
##############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
```

```
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

###############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
```

```
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

```
###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
```

```
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
```

```
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
```

```
  echo "Creates an AWS Identity and Access Management (IAM) policy."
  echo "  -n policy_name   The name of the IAM policy."
  echo "  -p policy_json -- The policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
  --query Policy.Arn)

local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
```

```
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
```

```
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
```

```
  echo "Deletes an WS Identity and Access Management (IAM) policy"
  echo "  -n policy_arn -- The name of the IAM policy arn."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy arn with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Policy arn:  $policy_arn"
iecho ""

response=$(aws iam delete-policy \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
  return 1
fi

iecho "delete-policy response:$response"
```

```
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an WS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
```

```
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an WS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
```

```
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user."
```

```
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name does not exist in the account."
  return 1
fi

response=$(aws iam delete-user \
  --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-user operation failed.$response"
```

```
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento dei comandi AWS CLI .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

C++

SDK per C++

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace AwsDoc {
```

```
    namespace IAM {

        //! Cleanup by deleting created entities.
        /*!
          \sa DeleteCreatedEntities
          \param client: IAM client.
          \param role: IAM role.
          \param user: IAM user.
          \param policy: IAM policy.
        */
        static bool DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                          const Aws::IAM::Model::Role &role,
                                          const Aws::IAM::Model::User &user,
                                          const Aws::IAM::Model::Policy &policy);
    }

    static const int LIST_BUCKETS_WAIT_SEC = 20;

    static const char ALLOCATION_TAG[] = "example_code";
}

//! Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
// "IAM access" permissions are needed to run this code.
// "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
//    create a custom policy).
/*!
  \sa iamCreateUserAssumeRoleScenario
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
*/
bool AwsDoc::IAM::iamCreateUserAssumeRoleScenario(
        const Aws::Client::ClientConfiguration &clientConfig) {

    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::User user;
    Aws::IAM::Model::Role role;
    Aws::IAM::Model::Policy policy;

    // 1. Create a user.
    {
        Aws::IAM::Model::CreateUserRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
```

```
        Aws::String userName = "iam-demo-user-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetUserName(userName);

        Aws::IAM::Model::CreateUserOutcome outcome = client.CreateUser(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Error creating IAM user " << userName << ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }
        else {
            std::cout << "Successfully created IAM user " << userName <<
std::endl;
        }

        user = outcome.GetResult().GetUser();
    }

    // 2. Create a role.
    {
        // Get the IAM user for the current client in order to access its ARN.
        Aws::String iamUserArn;
        {
            Aws::IAM::Model::GetUserRequest request;
            Aws::IAM::Model::GetUserOutcome outcome = client.GetUser(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error getting Iam user. " <<
                          outcome.GetError().GetMessage() << std::endl;

                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }
            else {
                std::cout << "Successfully retrieved Iam user "
                          << outcome.GetResult().GetUser().GetUserName()
                          << std::endl;
            }

            iamUserArn = outcome.GetResult().GetUser().GetArn();
        }

        Aws::IAM::Model::CreateRoleRequest request;

        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
```

```
        Aws::String roleName = "iam-demo-role-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleName(roleName);

        // Build policy document for role.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");

        Aws::Utils::Document jsonPrincipal;
        jsonPrincipal.WithString("AWS", iamUserArn);
        jsonStatement.WithObject("Principal", jsonPrincipal);
        jsonStatement.WithString("Action", "sts:AssumeRole");
        jsonStatement.WithObject("Condition", Aws::Utils::Document());

        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Setting policy for role\n   "
                  << policyDocument.View().WriteCompact() << std::endl;

        // Set role policy document as JSON string.

    request.SetAssumeRolePolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating role. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a role with name " << roleName
                      << std::endl;
        }

        role = outcome.GetResult().GetRole();
    }
```

```
    // 3. Create an IAM policy.
    {
        Aws::IAM::Model::CreatePolicyRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String policyName = "iam-demo-policy-" +
                                 Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetPolicyName(policyName);

        // Build IAM policy document.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");
        jsonStatement.WithString("Action", "s3:ListAllMyBuckets");
        jsonStatement.WithString("Resource", "arn:aws:s3:::*");

        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Creating a policy.\n   " <<
policyDocument.View().WriteCompact()
                  << std::endl;

        // Set IAM policy document as JSON string.
        request.SetPolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreatePolicyOutcome outcome =
client.CreatePolicy(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a policy with name, " <<
policyName <<
                      "." << std::endl;
        }

        policy = outcome.GetResult().GetPolicy();
```

```
    }

    // 4. Assume the new role using the AWS Security Token Service (STS).
    Aws::STS::Model::Credentials credentials;
    {
        Aws::STS::STSClient stsClient(clientConfig);

        Aws::STS::Model::AssumeRoleRequest request;
        request.SetRoleArn(role.GetArn());
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleSessionName = "iam-demo-role-session-" +

Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleSessionName(roleSessionName);

        Aws::STS::Model::AssumeRoleOutcome assumeRoleOutcome;

        // Repeatedly call AssumeRole, because there is often a delay
        // before the role is available to be assumed.
        // Repeat at most 20 times when access is denied.
        int count = 0;
        while (true) {
            assumeRoleOutcome = stsClient.AssumeRole(request);
            if (!assumeRoleOutcome.IsSuccess()) {
                if (count > 20 ||
                    assumeRoleOutcome.GetError().GetErrorType() !=
                    Aws::STS::STSErrors::ACCESS_DENIED) {
                    std::cerr << "Error assuming role after 20 tries. " <<
                              assumeRoleOutcome.GetError().GetMessage() <<
std::endl;

                    DeleteCreatedEntities(client, role, user, policy);
                    return false;
                }
                std::this_thread::sleep_for(std::chrono::seconds(1));
            }
            else {
                std::cout << "Successfully assumed the role after " << count
                          << " seconds." << std::endl;
                break;
            }
            count++;
        }
```

```
        credentials = assumeRoleOutcome.GetResult().GetCredentials();
    }


    // 5. List objects in the bucket (This should fail).
    {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if (listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
std::endl;
            }
            else {
                std::cout
                        << "Access to list buckets denied because privileges have
not been applied."
                        << std::endl;
            }
        }
        else {
            std::cerr
                    << "Successfully retrieved bucket lists when this should not
happen."
                    << std::endl;
        }
    }

    // 6. Attach the policy to the role.
    {
        Aws::IAM::Model::AttachRolePolicyRequest request;
        request.SetRoleName(role.GetRoleName());
        request.WithPolicyArn(policy.GetArn());

        Aws::IAM::Model::AttachRolePolicyOutcome outcome =
client.AttachRolePolicy(
```

```
                request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully attached the policy with name, "
                      << policy.GetPolicyName() <<
                      ", to the role, " << role.GetRoleName() << "." <<
std::endl;
        }
    }

    int count = 0;
    // 7. List objects in the bucket (this should succeed).
    // Repeatedly call ListBuckets, because there is often a delay
    // before the policy with ListBucket permissions has been applied to the
role.
    // Repeat at most LIST_BUCKETS_WAIT_SEC times when access is denied.
    while (true) {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if ((count > LIST_BUCKETS_WAIT_SEC) ||
                listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets after " <<
LIST_BUCKETS_WAIT_SEC << " seconds. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
std::endl;
                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }

            std::this_thread::sleep_for(std::chrono::seconds(1));
```

```
        }
        else {

            std::cout << "Successfully retrieved bucket lists after " << count
                      << " seconds." << std::endl;
            break;
        }
        count++;
    }

    // 8. Delete all the created resources.
    return DeleteCreatedEntities(client, role, user, policy);
}

bool AwsDoc::IAM::DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                        const Aws::IAM::Model::Role &role,
                                        const Aws::IAM::Model::User &user,
                                        const Aws::IAM::Model::Policy &policy) {
    bool result = true;
    if (policy.ArnHasBeenSet()) {
        // Detach the policy from the role.
        {
            Aws::IAM::Model::DetachRolePolicyRequest request;
            request.SetPolicyArn(policy.GetArn());
            request.SetRoleName(role.GetRoleName());

            Aws::IAM::Model::DetachRolePolicyOutcome outcome =
 client.DetachRolePolicy(
                    request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error Detaching policy from roles. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully detached the policy with arn "
                          << policy.GetArn()
                          << " from role " << role.GetRoleName() << "." <<
 std::endl;
            }
        }

        // Delete the policy.
        {
```

```
            Aws::IAM::Model::DeletePolicyRequest request;
            request.WithPolicyArn(policy.GetArn());

            Aws::IAM::Model::DeletePolicyOutcome outcome =
client.DeletePolicy(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error deleting policy. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully deleted the policy with arn "
                          << policy.GetArn() << std::endl;
            }
        }

    }

    if (role.RoleIdHasBeenSet()) {
        // Delete the role.
        Aws::IAM::Model::DeleteRoleRequest request;
        request.SetRoleName(role.GetRoleName());

        Aws::IAM::Model::DeleteRoleOutcome outcome = client.DeleteRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting role. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the role with name "
                      << role.GetRoleName() << std::endl;
        }
    }

    if (user.ArnHasBeenSet()) {
        // Delete the user.
        Aws::IAM::Model::DeleteUserRequest request;
        request.WithUserName(user.GetUserName());

        Aws::IAM::Model::DeleteUserOutcome outcome = client.DeleteUser(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting user. " <<
                      outcome.GetError().GetMessage() << std::endl;
```

```
            result = false;
        }
        else {
            std::cout << "Successfully deleted the user with name "
                      << user.GetUserName() << std::endl;
        }
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for C++ .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
// AssumeRoleScenario shows you how to use the AWS Identity and Access Management
 (IAM)
// service to perform the following actions:
//
// 1. Create a user who has no permissions.
// 2. Create a role that grants permission to list Amazon Simple Storage Service
//    (Amazon S3) buckets for the account.
// 3. Add a policy to let the user assume the role.
// 4. Try and fail to list buckets without permissions.
// 5. Assume the role and list S3 buckets using temporary credentials.
// 6. Delete the policy, role, and user.
type AssumeRoleScenario struct {
 sdkConfig aws.Config
 accountWrapper actions.AccountWrapper
 policyWrapper actions.PolicyWrapper
 roleWrapper actions.RoleWrapper
 userWrapper actions.UserWrapper
 questioner demotools.IQuestioner
 helper IScenarioHelper
 isTestRun bool
}

// NewAssumeRoleScenario constructs an AssumeRoleScenario instance from a
 configuration.
// It uses the specified config to get an IAM client and create wrappers for the
 actions
// used in the scenario.
func NewAssumeRoleScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
  helper IScenarioHelper) AssumeRoleScenario {
 iamClient := iam.NewFromConfig(sdkConfig)
 return AssumeRoleScenario{
  sdkConfig:   sdkConfig,
  accountWrapper: actions.AccountWrapper{IamClient: iamClient},
  policyWrapper:  actions.PolicyWrapper{IamClient: iamClient},
  roleWrapper:    actions.RoleWrapper{IamClient: iamClient},
  userWrapper:    actions.UserWrapper{IamClient: iamClient},
  questioner:     questioner,
  helper:         helper,
 }
}
```

```
// addTestOptions appends the API options specified in the original configuration
 to
// another configuration. This is used to attach the middleware stubber to
 clients
// that are constructed during the scenario, which is needed for unit testing.
func (scenario AssumeRoleScenario) addTestOptions(scenarioConfig *aws.Config) {
 if scenario.isTestRun {
  scenarioConfig.APIOptions = append(scenarioConfig.APIOptions,
 scenario.sdkConfig.APIOptions...)
 }
}

// Run runs the interactive scenario.
func (scenario AssumeRoleScenario) Run() {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong with the demo.\n")
   log.Println(r)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the AWS Identity and Access Management (IAM) assume role
 demo.")
 log.Println(strings.Repeat("-", 88))

 user := scenario.CreateUser()
 accessKey := scenario.CreateAccessKey(user)
 role := scenario.CreateRoleAndPolicies(user)
 noPermsConfig := scenario.ListBucketsWithoutPermissions(accessKey)
 scenario.ListBucketsWithAssumedRole(noPermsConfig, role)
 scenario.Cleanup(user, role)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateUser creates a new IAM user. This user has no permissions.
func (scenario AssumeRoleScenario) CreateUser() *types.User {
 log.Println("Let's create an example user with no permissions.")
 userName := scenario.questioner.Ask("Enter a name for the example user:",
 demotools.NotEmpty{})
```

```
 user, err := scenario.userWrapper.GetUser(userName)
 if err != nil {
  panic(err)
 }
 if user == nil {
  user, err = scenario.userWrapper.CreateUser(userName)
  if err != nil {
   panic(err)
  }
  log.Printf("Created user %v.\n", *user.UserName)
 } else {
  log.Printf("User %v already exists.\n", *user.UserName)
 }
 log.Println(strings.Repeat("-", 88))
 return user
}

// CreateAccessKey creates an access key for the user.
func (scenario AssumeRoleScenario) CreateAccessKey(user *types.User)
 *types.AccessKey {
 accessKey, err := scenario.userWrapper.CreateAccessKeyPair(*user.UserName)
 if err != nil {
  panic(err)
 }
 log.Printf("Created access key %v for your user.", *accessKey.AccessKeyId)
 log.Println("Waiting a few seconds for your user to be ready...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return accessKey
}

// CreateRoleAndPolicies creates a policy that grants permission to list S3
 buckets for
// the current account and attaches the policy to a newly created role. It also
 adds an
// inline policy to the specified user that grants the user permission to assume
 the role.
func (scenario AssumeRoleScenario) CreateRoleAndPolicies(user *types.User)
 *types.Role {
 log.Println("Let's create a role and policy that grant permission to list S3
 buckets.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 listBucketsRole, err :=
 scenario.roleWrapper.CreateRole(scenario.helper.GetName(), *user.Arn)
```

```
 if err != nil {panic(err)}
 log.Printf("Created role %v.\n", *listBucketsRole.RoleName)
 listBucketsPolicy, err := scenario.policyWrapper.CreatePolicy(
  scenario.helper.GetName(), []string{"s3:ListAllMyBuckets"}, "arn:aws:s3:::*")
 if err != nil {panic(err)}
 log.Printf("Created policy %v.\n", *listBucketsPolicy.PolicyName)
 err = scenario.roleWrapper.AttachRolePolicy(*listBucketsPolicy.Arn,
 *listBucketsRole.RoleName)
 if err != nil {panic(err)}
 log.Printf("Attached policy %v to role %v.\n", *listBucketsPolicy.PolicyName,
  *listBucketsRole.RoleName)
 err = scenario.userWrapper.CreateUserPolicy(*user.UserName,
 scenario.helper.GetName(),
  []string{"sts:AssumeRole"}, *listBucketsRole.Arn)
 if err != nil {panic(err)}
 log.Printf("Created an inline policy for user %v that lets the user assume the
 role.\n",
  *user.UserName)
 log.Println("Let's give AWS a few seconds to propagate these new resources and
 connections...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return listBucketsRole
}

// ListBucketsWithoutPermissions creates an Amazon S3 client from the user's
 access key
// credentials and tries to list buckets for the account. Because the user does
 not have
// permission to perform this action, the action fails.
func (scenario AssumeRoleScenario) ListBucketsWithoutPermissions(accessKey
 *types.AccessKey) *aws.Config {
  log.Println("Let's try to list buckets without permissions. This should return
 an AccessDenied error.")
  scenario.questioner.Ask("Press Enter when you're ready.")
  noPermsConfig, err := config.LoadDefaultConfig(context.TODO(),
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *accessKey.AccessKeyId, *accessKey.SecretAccessKey, ""),
 ))
  if err != nil {panic(err)}

  // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&noPermsConfig)
```

```
  s3Client := s3.NewFromConfig(noPermsConfig)
  _, err = s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
  if err != nil {
   // The SDK for Go does not model the AccessDenied error, so check ErrorCode
 directly.
  var ae smithy.APIError
  if errors.As(err, &ae) {
   switch ae.ErrorCode() {
   case "AccessDenied":
    log.Println("Got AccessDenied error, which is the expected result because\n"
 +
     "the ListBuckets call was made without permissions.")
   default:
    log.Println("Expected AccessDenied, got something else.")
    panic(err)
   }
  }
  } else {
   log.Println("Expected AccessDenied error when calling ListBuckets without
 permissions,\n" +
    "but the call succeeded. Continuing the example anyway...")
 }
 log.Println(strings.Repeat("-", 88))
  return &noPermsConfig
}

// ListBucketsWithAssumedRole performs the following actions:
//
// 1. Creates an AWS Security Token Service (AWS STS) client from the config
 created from
//   the user's access key credentials.
// 2. Gets temporary credentials by assuming the role that grants permission to
 list the
//    buckets.
// 3. Creates an Amazon S3 client from the temporary credentials.
// 4. Lists buckets for the account. Because the temporary credentials are
 generated by
//    assuming the role that grants permission, the action succeeds.
func (scenario AssumeRoleScenario) ListBucketsWithAssumedRole(noPermsConfig
 *aws.Config, role *types.Role) {
 log.Println("Let's assume the role that grants permission to list buckets and
 try again.")
 scenario.questioner.Ask("Press Enter when you're ready.")
```

```
 stsClient := sts.NewFromConfig(*noPermsConfig)
 tempCredentials, err := stsClient.AssumeRole(context.TODO(),
 &sts.AssumeRoleInput{
  RoleArn:           role.Arn,
  RoleSessionName:   aws.String("AssumeRoleExampleSession"),
  DurationSeconds:   aws.Int32(900),
 })
 if err != nil {
  log.Printf("Couldn't assume role %v.\n", *role.RoleName)
  panic(err)
 }
 log.Printf("Assumed role %v, got temporary credentials.\n", *role.RoleName)
 assumeRoleConfig, err := config.LoadDefaultConfig(context.TODO(),
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *tempCredentials.Credentials.AccessKeyId,
   *tempCredentials.Credentials.SecretAccessKey,
   *tempCredentials.Credentials.SessionToken),
  ),
 )
 if err != nil {panic(err)}

 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&assumeRoleConfig)

 s3Client := s3.NewFromConfig(assumeRoleConfig)
 result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
 if err != nil {
  log.Println("Couldn't list buckets with assumed role credentials.")
  panic(err)
 }
 log.Println("Successfully called ListBuckets with assumed role credentials, \n"
 +
  "here are some of them:")
 for i := 0; i < len(result.Buckets) && i < 5; i++ {
  log.Printf("\t%v\n", *result.Buckets[i].Name)
 }
 log.Println(strings.Repeat("-", 88))
}

// Cleanup deletes all resources created for the scenario.
func (scenario AssumeRoleScenario) Cleanup(user *types.User, role *types.Role) {
 if scenario.questioner.AskBool(
  "Do you want to delete the resources created for this example? (y/n)", "y",
```

```
 ) {
   policies, err := scenario.roleWrapper.ListAttachedRolePolicies(*role.RoleName)
   if err != nil {panic(err)}
   for _, policy := range policies {
    err = scenario.roleWrapper.DetachRolePolicy(*role.RoleName,
 *policy.PolicyArn)
    if err != nil {panic(err)}
    err = scenario.policyWrapper.DeletePolicy(*policy.PolicyArn)
    if err != nil {panic(err)}
    log.Printf("Detached policy %v from role %v and deleted the policy.\n",
     *policy.PolicyName, *role.RoleName)
   }
   err = scenario.roleWrapper.DeleteRole(*role.RoleName)
   if err != nil {panic(err)}
   log.Printf("Deleted role %v.\n", *role.RoleName)

   userPols, err := scenario.userWrapper.ListUserPolicies(*user.UserName)
   if err != nil {panic(err)}
   for _, userPol := range userPols {
    err = scenario.userWrapper.DeleteUserPolicy(*user.UserName, userPol)
    if err != nil {panic(err)}
    log.Printf("Deleted policy %v from user %v.\n", userPol, *user.UserName)
   }
   keys, err := scenario.userWrapper.ListAccessKeys(*user.UserName)
   if err != nil {panic(err)}
   for _, key := range keys {
    err = scenario.userWrapper.DeleteAccessKey(*user.UserName, *key.AccessKeyId)
    if err != nil {panic(err)}
    log.Printf("Deleted access key %v from user %v.\n", *key.AccessKeyId,
 *user.UserName)
   }
   err = scenario.userWrapper.DeleteUser(*user.UserName)
   if err != nil {panic(err)}
   log.Printf("Deleted user %v.\n", *user.UserName)
   log.Println(strings.Repeat("-", 88))
 }

}
```

Definisci una struttura che racchiude le azioni dell'account.

```
// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}



// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy() (*types.PasswordPolicy,
 error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(context.TODO(),
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}



// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders() ([]types.SAMLProviderListEntry,
 error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(context.TODO(),
 &iam.ListSAMLProvidersInput{})
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

Definisci una struttura che racchiude le azioni della policy.

```
// PolicyDocument defines a policy document as a Go struct that can be serialized
// to JSON.
type PolicyDocument struct {
 Version string
 Statement []PolicyStatement
}

// PolicyStatement defines a statement in a policy document.
type PolicyStatement struct {
 Effect string
 Action []string
 Principal map[string]string `json:",omitempty"`
 Resource *string `json:",omitempty"`
}



// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(maxPolicies int32) ([]types.Policy,
 error) {
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(context.TODO(),
 &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
```

```
  policies = result.Policies
 }
 return policies, err
}



// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(policyName string, actions []string,
  resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(resourceArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(context.TODO(),
 &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}
```

```
// GetPolicy gets data about a policy.
func (wrapper PolicyWrapper) GetPolicy(policyArn string) (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(context.TODO(), &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(policyArn string) error {
 _, err := wrapper.IamClient.DeletePolicy(context.TODO(), &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni del ruolo.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
func (wrapper RoleWrapper) ListRoles(maxRoles int32) ([]types.Role, error) {
```

```
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(context.TODO(),
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
 return roles, err
}



// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(roleName string, trustedUserArn string)
 (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action: []string{"sts:AssumeRole"},
  }},
 }
 policyBytes, err := json.Marshal(trustPolicy)
 if err != nil {
  log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
 trustedUserArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreateRole(context.TODO(),
 &iam.CreateRoleInput{
  AssumeRolePolicyDocument: aws.String(string(policyBytes)),
  RoleName:                 aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
```

```
 } else {
  role = result.Role
 }
 return role, err
}



// GetRole gets data about a role.
func (wrapper RoleWrapper) GetRole(roleName string) (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.GetRole(context.TODO(),
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(serviceName string,
 description string) (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(context.TODO(),
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}
```

```
// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(context.TODO(),
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(policyArn string, roleName string)
 error {
 _, err := wrapper.IamClient.AttachRolePolicy(context.TODO(),
 &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}



// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(roleName string)
 ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(context.TODO(),
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
```

```
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}



// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(roleName string, policyArn string)
 error {
 _, err := wrapper.IamClient.DetachRolePolicy(context.TODO(),
 &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(roleName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(context.TODO(),
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteRole deletes a role. All attached policies must be detached before a
```

```
// role can be deleted.
func (wrapper RoleWrapper) DeleteRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteRole(context.TODO(), &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni dell'utente.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(maxUsers int32) ([]types.User, error) {
 var users []types.User
 result, err := wrapper.IamClient.ListUsers(context.TODO(), &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
 return users, err
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(userName string) (*types.User, error) {
```

```
 var user *types.User
 result, err := wrapper.IamClient.GetUser(context.TODO(), &iam.GetUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
  user = result.User
 }
 return user, err
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(userName string) (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(context.TODO(),
 &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
 }
 return user, err
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
```

```
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(userName string, policyName string,
 actions []string,
  roleArn string) error {
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(roleArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
 err)
  return err
 }
 _, err = wrapper.IamClient.PutUserPolicy(context.TODO(),
 &iam.PutUserPolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
  UserName:       aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(userName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(context.TODO(),
 &iam.ListUserPoliciesInput{
  UserName: aws.String(userName),
 })
 if err != nil {
```

```
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(userName string, policyName string)
 error {
 _, err := wrapper.IamClient.DeleteUserPolicy(context.TODO(),
 &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(userName string) error {
 _, err := wrapper.IamClient.DeleteUser(context.TODO(), &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}



// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
```

```
func (wrapper UserWrapper) CreateAccessKeyPair(userName string)
 (*types.AccessKey, error) {
 var key *types.AccessKey
 result, err := wrapper.IamClient.CreateAccessKey(context.TODO(),
 &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}



// DeleteAccessKey deletes an access key from a user.
func (wrapper UserWrapper) DeleteAccessKey(userName string, keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(context.TODO(),
 &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(context.TODO(),
 &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
```

```
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
/*
```

```
  To run this Java V2 code example, set up your development environment,
 including your credentials.

  For information, see this documentation topic:

  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html

  This example performs these operations:

  1. Creates a user that has no permissions.
  2. Creates a role and policy that grants Amazon S3 permissions.
  3. Creates a role.
  4. Grants the user permissions.
  5. Gets temporary credentials by assuming the role.  Creates an Amazon S3
 Service client object with the temporary credentials.
  6. Deletes the resources.
 */

public class IAMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static String userArn;

    public static void main(String[] args) throws Exception {

        final String usage = """

                Usage:
                    <username> <policyName> <roleName> <roleSessionName>
 <bucketName>\s
```

```
                Where:
                    username - The name of the IAM user to create.\s
                    policyName - The name of the policy to create.\s
                    roleName - The name of the role to create.\s
                    roleSessionName - The name of the session required for the
assumeRole operation.\s
                    bucketName - The name of the Amazon S3 bucket from which
objects are read.\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        String policyName = args[1];
        String roleName = args[2];
        String roleSessionName = args[3];
        String bucketName = args[4];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the AWS IAM example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 1. Create the IAM user.");
        User createUser = createIAMUser(iam, userName);

        System.out.println(DASHES);
        userArn = createUser.arn();

        AccessKey myKey = createIAMAccessKey(iam, userName);
        String accessKey = myKey.accessKeyId();
        String secretKey = myKey.secretAccessKey();
        String assumeRolePolicyDocument = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
```

```
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                " \"AWS\": \"" + userArn + "\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        System.out.println(assumeRolePolicyDocument);
        System.out.println(userName + " was successfully created.");
        System.out.println(DASHES);
        System.out.println("2. Creates a policy.");
        String polArn = createIAMPolicy(iam, policyName);
        System.out.println("The policy " + polArn + " was successfully
created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Creates a role.");
        TimeUnit.SECONDS.sleep(30);
        String roleArn = createIAMRole(iam, roleName, assumeRolePolicyDocument);
        System.out.println(roleArn + " was successfully created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Grants the user permissions.");
        attachIAMRolePolicy(iam, roleName, polArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Wait for 30 secs so the resource is available");
        TimeUnit.SECONDS.sleep(30);
        System.out.println("5. Gets temporary credentials by assuming the
role.");
        System.out.println("Perform an Amazon S3 Service operation using the
temporary credentials.");
        assumeRole(roleArn, roleSessionName, bucketName, accessKey, secretKey);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6 Getting ready to delete the AWS resources");
        deleteKey(iam, userName, accessKey);
        deleteRole(iam, roleName, polArn);
        deleteIAMUser(iam, userName);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("This IAM Scenario has successfully completed");
        System.out.println(DASHES);
    }

    public static AccessKey createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();

            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static User createIAMUser(IamClient iam, String username) {
        try {
            // Create an IamWaiter object
            IamWaiter iamWaiter = iam.waiter();
            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            // Wait until the user is created.
            CreateUserResponse response = iam.createUser(request);
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
iamWaiter.waitUntilUserExists(userRequest);

waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user();

        } catch (IamException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static String createIAMRole(IamClient iam, String rolename, String
json) {

        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(json)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            System.out.println("The ARN of the role is " +
response.role().arn());
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();
            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument).build();

            CreatePolicyResponse response = iam.createPolicy(request);
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();

            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
iamWaiter.waitUntilPolicyExists(polRequest);
```

```
waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
           return response.policy().arn();

       } catch (IamException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return "";
   }

   public static void attachIAMRolePolicy(IamClient iam, String roleName, String
policyArn) {
       try {
           ListAttachedRolePoliciesRequest request =
ListAttachedRolePoliciesRequest.builder()
                   .roleName(roleName)
                   .build();

           ListAttachedRolePoliciesResponse response =
iam.listAttachedRolePolicies(request);
           List<AttachedPolicy> attachedPolicies = response.attachedPolicies();
           String polArn;
           for (AttachedPolicy policy : attachedPolicies) {
               polArn = policy.policyArn();
               if (polArn.compareTo(policyArn) == 0) {
                   System.out.println(roleName + " policy is already attached to
this role.");
                   return;
               }
           }

           AttachRolePolicyRequest attachRequest =
AttachRolePolicyRequest.builder()
                   .roleName(roleName)
                   .policyArn(policyArn)
                   .build();

           iam.attachRolePolicy(attachRequest);
           System.out.println("Successfully attached policy " + policyArn + " to
role " + roleName);

       } catch (IamException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }

    // Invoke an Amazon S3 operation using the Assumed Role.
    public static void assumeRole(String roleArn, String roleSessionName, String
bucketName, String keyVal,
            String keySecret) {

        // Use the creds of the new IAM user that was created in this code
example.
        AwsBasicCredentials credentials = AwsBasicCredentials.create(keyVal,
keySecret);
        StsClient stsClient = StsClient.builder()
                .region(Region.US_EAST_1)

.credentialsProvider(StaticCredentialsProvider.create(credentials))
                .build();

        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();
            String key = myCreds.accessKeyId();
            String secKey = myCreds.secretAccessKey();
            String secToken = myCreds.sessionToken();

            // List all objects in an Amazon S3 bucket using the temp creds
retrieved by
            // invoking assumeRole.
            Region region = Region.US_EAST_1;
            S3Client s3 = S3Client.builder()
                    .credentialsProvider(

StaticCredentialsProvider.create(AwsSessionCredentials.create(key, secKey,
secToken)))
                    .region(region)
                    .build();

            System.out.println("Created a S3Client using temp credentials.");
```

```
            System.out.println("Listing objects in " + bucketName);
            ListObjectsRequest listObjects = ListObjectsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListObjectsResponse res = s3.listObjects(listObjects);
            List<S3Object> objects = res.contents();
            for (S3Object myValue : objects) {
                System.out.println("The name of the key is " + myValue.key());
                System.out.println("The owner is " + myValue.owner());
            }

        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void deleteRole(IamClient iam, String roleName, String polArn)
{

        try {
            // First the policy needs to be detached.
            DetachRolePolicyRequest rolePolicyRequest =
DetachRolePolicyRequest.builder()
                    .policyArn(polArn)
                    .roleName(roleName)
                    .build();

            iam.detachRolePolicy(rolePolicyRequest);

            // Delete the policy.
            DeletePolicyRequest request = DeletePolicyRequest.builder()
                    .policyArn(polArn)
                    .build();

            iam.deletePolicy(request);
            System.out.println("*** Successfully deleted " + polArn);

            // Delete the role.
            DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                    .roleName(roleName)
                    .build();
```

```
            iam.deleteRole(roleRequest);
            System.out.println("*** Successfully deleted " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteKey(IamClient iam, String username, String
 accessKey) {
        try {
            DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                    .accessKeyId(accessKey)
                    .userName(username)
                    .build();

            iam.deleteAccessKey(request);
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("*** Successfully deleted " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import {
  CreateUserCommand,
  GetUserCommand,
  CreateAccessKeyCommand,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
```

```
  DeleteAccessKeyCommand,
  DeleteUserCommand,
  DeleteRoleCommand,
  DeletePolicyCommand,
  DetachRolePolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
import { AssumeRoleCommand, STSClient } from "@aws-sdk/client-sts";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { ScenarioInput } from "@aws-doc-sdk-examples/lib/scenario/index.js";

// Set the parameters.
const iamClient = new IAMClient({});
const userName = "test_name";
const policyName = "test_policy";
const roleName = "test_role";

/**
 * Create a new IAM user. If the user already exists, give
 * the option to delete and re-create it.
 * @param {string} name
 */
export const createUser = async (name, confirmAll = false) => {
  try {
    const { User } = await iamClient.send(
      new GetUserCommand({ UserName: name }),
    );
    const input = new ScenarioInput(
      "deleteUser",
      "Do you want to delete and remake this user?",
      { type: "confirm" },
    );
    const deleteUser = await input.handle({}, { confirmAll });
    // If the user exists, and you want to delete it, delete the user
    // and then create it again.
    if (deleteUser) {
      await iamClient.send(new DeleteUserCommand({ UserName: User.UserName }));
      await iamClient.send(new CreateUserCommand({ UserName: name }));
    } else {
      console.warn(
        `${name} already exists. The scenario may not work as expected.`,
      );
      return User;
```

```
    }
  } catch (caught) {
    // If there is no user by that name, create one.
    if (caught instanceof Error && caught.name === "NoSuchEntityException") {
      const { User } = await iamClient.send(
        new CreateUserCommand({ UserName: name }),
      );
      return User;
    } else {
      throw caught;
    }
  }
};

export const main = async (confirmAll = false) => {
  // Create a user. The user has no permissions by default.
  const User = await createUser(userName, confirmAll);

  if (!User) {
    throw new Error("User not created");
  }

  // Create an access key. This key is used to authenticate the new user to
  // Amazon Simple Storage Service (Amazon S3) and AWS Security Token Service
 (AWS STS).
  // It's not best practice to use access keys. For more information, see
 https://aws.amazon.com/iam/resources/best-practices/.
  const createAccessKeyResponse = await iamClient.send(
    new CreateAccessKeyCommand({ UserName: userName }),
  );

  if (
    !createAccessKeyResponse.AccessKey?.AccessKeyId ||
    !createAccessKeyResponse.AccessKey?.SecretAccessKey
  ) {
    throw new Error("Access key not created");
  }

  const {
    AccessKey: { AccessKeyId, SecretAccessKey },
  } = createAccessKeyResponse;

  let s3Client = new S3Client({
    credentials: {
```

```
      accessKeyId: AccessKeyId,
      secretAccessKey: SecretAccessKey,
    },
  });

  // Retry the list buckets operation until it succeeds. InvalidAccessKeyId is
  // thrown while the user and access keys are still stabilizing.
  await retry({ intervalInMs: 1000, maxRetries: 300 }, async () => {
    try {
      return await listBuckets(s3Client);
    } catch (err) {
      if (err instanceof Error && err.name === "InvalidAccessKeyId") {
        throw err;
      }
    }
  });

  // Retry the create role operation until it succeeds. A MalformedPolicyDocument
error
  // is thrown while the user and access keys are still stabilizing.
  const { Role } = await retry(
    {
      intervalInMs: 2000,
      maxRetries: 60,
    },
    () =>
      iamClient.send(
        new CreateRoleCommand({
          AssumeRolePolicyDocument: JSON.stringify({
            Version: "2012-10-17",
            Statement: [
              {
                Effect: "Allow",
                Principal: {
                  // Allow the previously created user to assume this role.
                  AWS: User.Arn,
                },
                Action: "sts:AssumeRole",
              },
            ],
          }),
          RoleName: roleName,
        }),
      ),
```

```
);

if (!Role) {
  throw new Error("Role not created");
}

// Create a policy that allows the user to list S3 buckets.
const { Policy: listBucketPolicy } = await iamClient.send(
  new CreatePolicyCommand({
    PolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Action: ["s3:ListAllMyBuckets"],
          Resource: "*",
        },
      ],
    }),
    PolicyName: policyName,
  }),
);

if (!listBucketPolicy) {
  throw new Error("Policy not created");
}

// Attach the policy granting the 's3:ListAllMyBuckets' action to the role.
await iamClient.send(
  new AttachRolePolicyCommand({
    PolicyArn: listBucketPolicy.Arn,
    RoleName: Role.RoleName,
  }),
);

// Assume the role.
const stsClient = new STSClient({
  credentials: {
    accessKeyId: AccessKeyId,
    secretAccessKey: SecretAccessKey,
  },
});

// Retry the assume role operation until it succeeds.
```

```
  const { Credentials } = await retry(
    { intervalInMs: 2000, maxRetries: 60 },
    () =>
      stsClient.send(
        new AssumeRoleCommand({
          RoleArn: Role.Arn,
          RoleSessionName: `iamBasicScenarioSession-${Math.floor(
            Math.random() * 1000000,
          )}`,
          DurationSeconds: 900,
        }),
      ),
  );

  if (!Credentials?.AccessKeyId || !Credentials?.SecretAccessKey) {
    throw new Error("Credentials not created");
  }

  s3Client = new S3Client({
    credentials: {
      accessKeyId: Credentials.AccessKeyId,
      secretAccessKey: Credentials.SecretAccessKey,
      sessionToken: Credentials.SessionToken,
    },
  });

  // List the S3 buckets again.
  // Retry the list buckets operation until it succeeds. AccessDenied might
  // be thrown while the role policy is still stabilizing.
  await retry({ intervalInMs: 2000, maxRetries: 60 }, () =>
    listBuckets(s3Client),
  );

  // Clean up.
  await iamClient.send(
    new DetachRolePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeletePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
```

```
    }),
  );

  await iamClient.send(
    new DeleteRoleCommand({
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeleteAccessKeyCommand({
      UserName: userName,
      AccessKeyId,
    }),
  );

  await iamClient.send(
    new DeleteUserCommand({
      UserName: userName,
    }),
  );
};

/**
 *
 * @param {S3Client} s3Client
 */
const listBuckets = async (s3Client) => {
  const { Buckets } = await s3Client.send(new ListBucketsCommand({}));

  if (!Buckets) {
    throw new Error("Buckets not listed");
  }

  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - AttachRolePolitica

- CreateAccessChiave
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessChiave
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <username> <policyName> <roleName> <roleSessionName> <fileLocation>
 <bucketName>

    Where:
        username - The name of the IAM user to create.
        policyName - The name of the policy to create.
        roleName - The name of the role to create.
        roleSessionName - The name of the session required for the assumeRole
 operation.
```

```
        fileLocation - The file location to the JSON required to create the role
 (see Readme).
        bucketName - The name of the Amazon S3 bucket from which objects are
 read.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val userName = args[0]
    val policyName = args[1]
    val roleName = args[2]
    val roleSessionName = args[3]
    val fileLocation = args[4]
    val bucketName = args[5]

    createUser(userName)
    println("$userName was successfully created.")

    val polArn = createPolicy(policyName)
    println("The policy $polArn was successfully created.")

    val roleArn = createRole(roleName, fileLocation)
    println("$roleArn was successfully created.")
    attachRolePolicy(roleName, polArn)

    println("*** Wait for 1 MIN so the resource is available.")
    delay(60000)
    assumeGivenRole(roleArn, roleSessionName, bucketName)

    println("*** Getting ready to delete the AWS resources.")
    deleteRole(roleName, polArn)
    deleteUser(userName)
    println("This IAM Scenario has successfully completed.")
}

suspend fun createUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }
```

```
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}

suspend fun createPolicy(policyNameVal: String?): String {
    val policyDocumentValue: String =
        "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}"

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentValue
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}

suspend fun createRole(
    rolenameVal: String?,
    fileLocation: String?
): String? {
    val jsonObject = fileLocation?.let { readJsonSimpleDemo(it) } as JSONObject

    val request =
        CreateRoleRequest {
            roleName = rolenameVal
            assumeRolePolicyDocument = jsonObject.toJSONString()
            description = "Created using the AWS SDK for Kotlin"
```

```
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createRole(request)
        return response.role?.arn
    }
}

suspend fun attachRolePolicy(
    roleNameVal: String,
    policyArnVal: String
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkMyList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }

        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkMyList(
    attachedPolicies: List<AttachedPolicy>,
    policyArnVal: String
```

```
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
        }
    }
    return 0
}

suspend fun assumeGivenRole(
    roleArnVal: String?,
    roleSessionNameVal: String?,
    bucketName: String
) {
    val stsClient =
        StsClient {
            region = "us-east-1"
        }

    val roleRequest =
        AssumeRoleRequest {
            roleArn = roleArnVal
            roleSessionName = roleSessionNameVal
        }

    val roleResponse = stsClient.assumeRole(roleRequest)
    val myCreds = roleResponse.credentials
    val key = myCreds?.accessKeyId
    val secKey = myCreds?.secretAccessKey
    val secToken = myCreds?.sessionToken

    val staticCredentials =
        StaticCredentialsProvider {
            accessKeyId = key
            secretAccessKey = secKey
            sessionToken = secToken
        }

    // List all objects in an Amazon S3 bucket using the temp creds.
    val s3 =
        S3Client {
```

```
            credentialsProvider = staticCredentials
            region = "us-east-1"
        }

    println("Created a S3Client using temp credentials.")
    println("Listing objects in $bucketName")

    val listObjects =
        ListObjectsRequest {
            bucket = bucketName
        }

    val response = s3.listObjects(listObjects)
    response.contents?.forEach { myObject ->
        println("The name of the key is ${myObject.key}")
        println("The owner is ${myObject.owner}")
    }
}

suspend fun deleteRole(
    roleNameVal: String,
    polArn: String
) {
    val iam = IamClient { region = "AWS_GLOBAL" }

    // First the policy needs to be detached.
    val rolePolicyRequest =
        DetachRolePolicyRequest {
            policyArn = polArn
            roleName = roleNameVal
        }

    iam.detachRolePolicy(rolePolicyRequest)

    // Delete the policy.
    val request =
        DeletePolicyRequest {
            policyArn = polArn
        }

    iam.deletePolicy(request)
    println("*** Successfully deleted $polArn")

    // Delete the role.
```

```
    val roleRequest =
        DeleteRoleRequest {
            roleName = roleNameVal
        }

    iam.deleteRole(roleRequest)
    println("*** Successfully deleted $roleNameVal")
}

suspend fun deleteUser(userNameVal: String) {
    val iam = IamClient { region = "AWS_GLOBAL" }
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }

    iam.deleteUser(request)
    println("*** Successfully deleted $userNameVal")
}

@Throws(java.lang.Exception::class)
fun readJsonSimpleDemo(filename: String): Any? {
    val reader = FileReader(filename)
    val jsonParser = JSONParser()
    return jsonParser.parse(reader)
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser

- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

PHP

SDK per PHP

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace Iam\Basics;

require 'vendor/autoload.php';

use Aws\Credentials\Credentials;
use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;
use Iam\IAMService;

echo("\n");
echo("--------------------------------------\n");
print("Welcome to the IAM getting started demo using PHP!\n");
echo("--------------------------------------\n");

$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";

$key = $service->createAccessKey($user['UserName']);
$assumeRolePolicyDocument = "{
               \"Version\": \"2012-10-17\",
               \"Statement\": [{
                   \"Effect\": \"Allow\",
                   \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
```

```
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

$inlinePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{$assumeRoleRole['Arn']}\"}]
}";
$inlinePolicy = $service->createUserPolicy("iam_demo_inline_policy_$uuid",
 $inlinePolicyDocument, $user['UserName']);
//First, fail to list the buckets with the user
$credentials = new Credentials($key['AccessKeyId'], $key['SecretAccessKey']);
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
try {
    $s3Client->listBuckets([
    ]);
    echo "this should not run";
} catch (S3Exception $exception) {
    echo "successfully failed!\n";
}

$stsClient = new StsClient(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
```

```
sleep(10);
$assumedRole = $stsClient->assumeRole([
    'RoleArn' => $assumeRoleRole['Arn'],
    'RoleSessionName' => "DemoAssumeRoleSession_$uuid",
]);
$assumedCredentials = [
    'key' => $assumedRole['Credentials']['AccessKeyId'],
    'secret' => $assumedRole['Credentials']['SecretAccessKey'],
    'token' => $assumedRole['Credentials']['SessionToken'],
];
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $assumedCredentials]);
try {
    $s3Client->listBuckets([]);
    echo "this should now run!\n";
} catch (S3Exception $exception) {
    echo "this should now not fail\n";
}

$service->detachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);
$deletePolicy = $service->deletePolicy($listAllBucketsPolicy['Arn']);
echo "Delete policy: {$listAllBucketsPolicy['PolicyName']}\n";
$deletedRole = $service->deleteRole($assumeRoleRole['Arn']);
echo "Deleted role: {$assumeRoleRole['RoleName']}\n";
$deletedKey = $service->deleteAccessKey($key['AccessKeyId'], $user['UserName']);
$deletedUser = $service->deleteUser($user['UserName']);
echo "Delete user: {$user['UserName']}\n";
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for PHP .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy

- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import json
import sys
import time
from uuid import uuid4

import boto3
from botocore.exceptions import ClientError


def progress_bar(seconds):
    """Shows a simple progress bar in the command window."""
    for _ in range(seconds):
        time.sleep(1)
        print(".", end="")
        sys.stdout.flush()
    print()


def setup(iam_resource):
```

```
    """
    Creates a new user with no permissions.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
    Creates an inline policy for the user that lets the user assume the role.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
resource
                         that has permissions to create users, roles, and
policies
                         in the account.
    :return: The newly created user, user key, and role.
    """
    try:
        user = iam_resource.create_user(UserName=f"demo-user-{uuid4()}")
        print(f"Created user {user.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a user for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user_key = user.create_access_key_pair()
        print(f"Created access key pair for user.")
    except ClientError as error:
        print(
            f"Couldn't create access keys for user {user.name}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    try:
        role = iam_resource.create_role(
            RoleName=f"demo-role-{uuid4()}",
            AssumeRolePolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
```

```
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {"AWS": user.arn},
                            "Action": "sts:AssumeRole",
                        }
                    ],
                }
            ),
        )
        print(f"Created role {role.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a role for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        policy = iam_resource.create_policy(
            PolicyName=f"demo-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "s3:ListAllMyBuckets",
                            "Resource": "arn:aws:s3:::*",
                        }
                    ],
                }
            ),
        )
        role.attach_policy(PolicyArn=policy.arn)
        print(f"Created policy {policy.policy_name} and attached it to the
role.")
    except ClientError as error:
        print(
            f"Couldn't create a policy and attach it to role {role.name}. Here's
why: "
            f"{error.response['Error']['Message']}"
        )
        raise
```

```
    try:
        user.create_policy(
            PolicyName=f"demo-user-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "sts:AssumeRole",
                            "Resource": role.arn,
                        }
                    ],
                }
            ),
        )
        print(
            f"Created an inline policy for {user.name} that lets the user assume
 "
            f"the role."
        )
    except ClientError as error:
        print(
            f"Couldn't create an inline policy for user {user.name}. Here's why:
 "
            f"{error.response['Error']['Message']}"
        )
        raise

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, role


def show_access_denied_without_role(user_key):
    """
    Shows that listing buckets without first assuming the role is not allowed.

    :param user_key: The key of the user created during setup. This user does not
                     have permission to list buckets in the account.
    """
```

```
    print(f"Try to list buckets without first assuming the role.")
    s3_denied_resource = boto3.resource(
        "s3", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        for bucket in s3_denied_resource.buckets.all():
            print(bucket.name)
        raise RuntimeError("Expected to get AccessDenied error when listing
 buckets!")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Attempt to list buckets with no permissions: AccessDenied.")
        else:
            raise


def list_buckets_from_assumed_role(user_key, assume_role_arn, session_name):
    """
    Assumes a role that grants permission to list the Amazon S3 buckets in the
 account.
    Uses the temporary credentials from the role to list the buckets that are
 owned
    by the assumed role's account.

    :param user_key: The access key of a user that has permission to assume the
 role.
    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    """
    sts_client = boto3.client(
        "sts", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        response = sts_client.assume_role(
            RoleArn=assume_role_arn, RoleSessionName=session_name
        )
        temp_credentials = response["Credentials"]
        print(f"Assumed role {assume_role_arn} and got temporary credentials.")
    except ClientError as error:
        print(
            f"Couldn't assume role {assume_role_arn}. Here's why: "
```

```
            f"{error.response['Error']['Message']}"
        )
        raise

    # Create an S3 resource that can access the account with the temporary
 credentials.
    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
    print(f"Listing buckets for the assumed role's account:")
    try:
        for bucket in s3_resource.buckets.all():
            print(bucket.name)
    except ClientError as error:
        print(
            f"Couldn't list buckets for the account. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise




def teardown(user, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    try:
        for attached in role.attached_policies.all():
            policy_name = attached.policy_name
            role.detach_policy(PolicyArn=attached.arn)
            attached.delete()
            print(f"Detached and deleted {policy_name}.")
        role.delete()
        print(f"Deleted {role.name}.")
    except ClientError as error:
        print(
            "Couldn't detach policy, delete policy, or delete role. Here's why: "
```

```
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        for user_pol in user.policies.all():
            user_pol.delete()
            print("Deleted inline user policy.")
        for key in user.access_keys.all():
            key.delete()
            print("Deleted user's access key.")
        user.delete()
        print(f"Deleted {user.name}.")
    except ClientError as error:
        print(
            "Couldn't delete user policy or delete user. Here's why: "
            f"{error.response['Error']['Message']}"
        )


def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(f"Welcome to the IAM create user and assume role demo.")
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user = None
    role = None
    try:
        user, user_key, role = setup(iam_resource)
        print(f"Created {user.name} and {role.name}.")
        show_access_denied_without_role(user_key)
        list_buckets_from_assumed_role(user_key, role.arn,
 "AssumeRoleDemoSession")
    except Exception:
        print("Something went wrong!")
    finally:
        if user is not None and role is not None:
            teardown(user, role)
        print("Thanks for watching!")


if __name__ == "__main__":
    usage_demo()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Ruby

SDK per Ruby

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_client

  # @param [Aws::IAM::Client] iam_client: The AWS IAM client.
```

```
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Waits for the specified number of seconds.
  #
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts("Give AWS time to propagate resources...")
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_client.create_user(user_name: user_name).user
    @logger.info("Created demo user named #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Tried and failed to create demo user.")
    @logger.info("\t#{e.code}: #{e.message}")
    @logger.info("\nCan't continue the demo without a user!")
    raise
  else
    user
  end

  # Creates an access key for a user.
  #
  # @param user [Aws::IAM::User] The user that owns the key.
  # @return [Aws::IAM::AccessKeyPair] The newly created access key.
  def create_access_key_pair(user)
    user_key = @iam_client.create_access_key(user_name:
user.user_name).access_key
    @logger.info("Created accesskey pair for user #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create access keys for user #{user.user_name}.")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    user_key
  end
```

```
 # Creates a role that can be assumed by a user.
 #
 # @param role_name [String] The name to give the role.
 # @param user [Aws::IAM::User] The user who is granted permission to assume the
role.
 # @return [Aws::IAM::Role] The newly created role.
 def create_role(role_name, user)
   trust_policy = {
     Version: "2012-10-17",
     Statement: [{
                  Effect: "Allow",
                  Principal: {'AWS': user.arn},
                  Action: "sts:AssumeRole"
                }]
   }.to_json
   role = @iam_client.create_role(
     role_name: role_name,
     assume_role_policy_document: trust_policy
   ).role
   @logger.info("Created role #{role.role_name}.")
 rescue Aws::Errors::ServiceError => e
   @logger.info("Couldn't create a role for the demo. Here's why: ")
   @logger.info("\t#{e.code}: #{e.message}")
   raise
 else
   role
 end

 # Creates a policy that grants permission to list S3 buckets in the account,
and
 # then attaches the policy to a role.
 #
 # @param policy_name [String] The name to give the policy.
 # @param role [Aws::IAM::Role] The role that the policy is attached to.
 # @return [Aws::IAM::Policy] The newly created policy.
 def create_and_attach_role_policy(policy_name, role)
   policy_document = {
     Version: "2012-10-17",
     Statement: [{
                  Effect: "Allow",
                  Action: "s3:ListAllMyBuckets",
                  Resource: "arn:aws:s3:::*"
                }]
```

```
    }.to_json
    policy = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document
    ).policy
    @iam_client.attach_role_policy(
      role_name: role.role_name,
      policy_arn: policy.arn
    )
    @logger.info("Created policy #{policy.policy_name} and attached it to role
#{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a policy and attach it to role
#{role.role_name}. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an inline policy for a user that lets the user assume a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param user [Aws::IAM::User] The user that owns the policy.
  # @param role [Aws::IAM::Role] The role that can be assumed.
  # @return [Aws::IAM::UserPolicy] The newly created policy.
  def create_user_policy(policy_name, user, role)
    policy_document = {
      Version: "2012-10-17",
      Statement: [{
                   Effect: "Allow",
                   Action: "sts:AssumeRole",
                   Resource: role.arn
                 }]
    }.to_json
    @iam_client.put_user_policy(
      user_name: user.user_name,
      policy_name: policy_name,
      policy_document: policy_document
    )
    puts("Created an inline policy for #{user.user_name} that lets the user
assume role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create an inline policy for user #{user.user_name}.
Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
```

```
    raise
  end

  # Creates an Amazon S3 resource with specified credentials. This is separated
into a
  # factory function so that it can be mocked for unit testing.
  #
  # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
  def create_s3_resource(credentials)
    Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
  end

  # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
  # Because the resource uses credentials with limited access, it may not be able
to
  # list the S3 buckets.
  #
  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def list_buckets(s3_resource)
    count = 10
    s3_resource.buckets.each do |bucket|
      @logger.info "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
  rescue Aws::Errors::ServiceError => e
    if e.code == "AccessDenied"
      puts("Attempt to list buckets with no permissions: AccessDenied.")
    else
      @logger.info("Couldn't list buckets for the account. Here's why: ")
      @logger.info("\t#{e.code}: #{e.message}")
      raise
    end
  end

  # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
  # This is separated into a factory function so that it can be mocked for unit
testing.
  #
  # @param key_id [String] The ID of the access key used by the STS client.
  # @param key_secret [String] The secret part of the access key used by the STS
client.
```

```
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: "create-use-assume-role-scenario"
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end

 # Deletes a role. If the role has policies attached, they are detached and
 # deleted before the role is deleted.
 #
 # @param role_name [String] The name of the role to delete.
 def delete_role(role_name)
   @iam_client.list_attached_role_policies(role_name:
role_name).attached_policies.each do |policy|
     @iam_client.detach_role_policy(role_name: role_name, policy_arn:
policy.policy_arn)
     @iam_client.delete_policy(policy_arn: policy.policy_arn)
     @logger.info("Detached and deleted policy #{policy.policy_name}.")
   end
   @iam_client.delete_role({ role_name: role_name })
   @logger.info("Role deleted: #{role_name}.")
 rescue Aws::Errors::ServiceError => e
   @logger.info("Couldn't detach policies and delete role #{role.name}. Here's
why:")
   @logger.info("\t#{e.code}: #{e.message}")
   raise
 end
```

```
  # Deletes a user. If the user has inline policies or access keys, they are
 deleted
  # before the user is deleted.
  #
  # @param user [Aws::IAM::User] The user to delete.
  def delete_user(user_name)
    user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
    user.each do |key|
      @iam_client.delete_access_key({ access_key_id: key.access_key_id,
 user_name: user_name })
      @logger.info("Deleted access key #{key.access_key_id} for user
 '#{user_name}'.")
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the IAM create a user and assume a role demo!")
  puts("-" * 88)
  user = scenario.create_user("doc-example-user-#{Random.uuid}")
  user_key = scenario.create_access_key_pair(user)
  scenario.wait(10)
  role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
  scenario.create_and_attach_role_policy("doc-example-role-policy-
#{Random.uuid}", role)
  scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user,
 role)
  scenario.wait(10)
  puts("Try to list buckets with credentials for a user who has no permissions.")
  puts("Expect AccessDenied from this call.")
  scenario.list_buckets(
    scenario.create_s3_resource(Aws::Credentials.new(user_key.access_key_id,
 user_key.secret_access_key)))
  puts("Now, assume the role that grants permission.")
  temp_credentials = scenario.assume_role(
    role.arn, scenario.create_sts_client(user_key.access_key_id,
 user_key.secret_access_key))
```

```
  puts("Here are your buckets:")
  scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
  puts("Deleting role '#{role.role_name}' and attached policies.")
  scenario.delete_role(role.role_name)
  puts("Deleting user '#{user.user_name}', policies, and keys.")
  scenario.delete_user(user.user_name)
  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError => e
  puts("Something went wrong with the demo.")
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Client.new)) if
 $PROGRAM_NAME == __FILE__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Ruby .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_iam::Error as iamError;
use aws_sdk_iam::{config::Credentials as iamCredentials, config::Region, Client
 as iamClient};
use aws_sdk_s3::Client as s3Client;
use aws_sdk_sts::Client as stsClient;
use tokio::time::{sleep, Duration};
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), iamError> {
    let (client, uuid, list_all_buckets_policy_document, inline_policy_document)
 =
        initialize_variables().await;

    if let Err(e) = run_iam_operations(
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
    .await
    {
        println!("{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (iamClient, String, String, String) {
```

```
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));

    let shared_config =
 aws_config::from_env().region(region_provider).load().await;
    let client = iamClient::new(&shared_config);
    let uuid = Uuid::new_v4().to_string();

    let list_all_buckets_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
    }"
    .to_string();
    let inline_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{}\"}]
    }"
    .to_string();

    (
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
}

async fn run_iam_operations(
    client: iamClient,
    uuid: String,
    list_all_buckets_policy_document: String,
    inline_policy_document: String,
) -> Result<(), iamError> {
    let user = iam_service::create_user(&client, &format!("{}{}",
 "iam_demo_user_", uuid)).await?;
    println!("Created the user with the name: {}", user.user_name());
    let key = iam_service::create_access_key(&client, user.user_name()).await?;
```

```
    let assume_role_policy_document = "{
        \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }"
    .to_string()
    .replace("{}", user.arn());

    let assume_role_role = iam_service::create_role(
        &client,
        &format!("{}{}", "iam_demo_role_", uuid),
        &assume_role_policy_document,
    )
    .await?;
    println!("Created the role with the ARN: {}", assume_role_role.arn());

    let list_all_buckets_policy = iam_service::create_policy(
        &client,
        &format!("{}{}", "iam_demo_policy_", uuid),
        &list_all_buckets_policy_document,
    )
    .await?;
    println!(
        "Created policy: {}",
        list_all_buckets_policy.policy_name.as_ref().unwrap()
    );

    let attach_role_policy_result =
        iam_service::attach_role_policy(&client, &assume_role_role,
&list_all_buckets_policy)
            .await?;
    println!(
        "Attached the policy to the role: {:?}",
        attach_role_policy_result
    );

    let inline_policy_name = format!("{}{}", "iam_demo_inline_policy_", uuid);
    let inline_policy_document = inline_policy_document.replace("{}",
assume_role_role.arn());
    iam_service::create_user_policy(&client, &user, &inline_policy_name,
&inline_policy_document)
```

```
        .await?;
    println!("Created inline policy.");

    //First, fail to list the buckets with the user.
    let creds = iamCredentials::from_keys(key.access_key_id(),
key.secret_access_key(), None);
    let fail_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    println!("Fail config: {:?}", fail_config);
    let fail_client: s3Client = s3Client::new(&fail_config);
    match fail_client.list_buckets().send().await {
        Ok(e) => {
            println!("This should not run. {:?}", e);
        }
        Err(e) => {
            println!("Successfully failed with error: {:?}", e)
        }
    }

    let sts_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    let sts_client: stsClient = stsClient::new(&sts_config);
    sleep(Duration::from_secs(10)).await;
    let assumed_role = sts_client
        .assume_role()
        .role_arn(assume_role_role.arn())
        .role_session_name(&format!("{}{}", "iam_demo_assumerole_session_",
uuid))
        .send()
        .await;
    println!("Assumed role: {:?}", assumed_role);
    sleep(Duration::from_secs(10)).await;

    let assumed_credentials = iamCredentials::from_keys(
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
```

```
            .access_key_id(),
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .secret_access_key(),
        Some(
            assumed_role
                .as_ref()
                .unwrap()
                .credentials
                .as_ref()
                .unwrap()
                .session_token
                .clone(),
        ),
    );

    let succeed_config = aws_config::from_env()
        .credentials_provider(assumed_credentials)
        .load()
        .await;
    println!("succeed config: {:?}", succeed_config);
    let succeed_client: s3Client = s3Client::new(&succeed_config);
    sleep(Duration::from_secs(10)).await;
    match succeed_client.list_buckets().send().await {
        Ok(_) => {
            println!("This should now run successfully.")
        }
        Err(e) => {
            println!("This should not run. {:?}", e);
            panic!()
        }
    }

    //Clean up.
    iam_service::detach_role_policy(
        &client,
        assume_role_role.role_name(),
        list_all_buckets_policy.arn().unwrap_or_default(),
    )
    .await?;
```

```
    iam_service::delete_policy(&client, list_all_buckets_policy).await?;
    iam_service::delete_role(&client, &assume_role_role).await?;
    println!("Deleted role {}", assume_role_role.role_name());
    iam_service::delete_access_key(&client, &user, &key).await?;
    println!("Deleted key for {}", key.user_name());
    iam_service::delete_user_policy(&client, &user, &inline_policy_name).await?;
    println!("Deleted inline user policy: {}", inline_policy_name);
    iam_service::delete_user(&client, &user).await?;
    println!("Deleted user {}", user.user_name());

    Ok(())
}
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Rust.
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2

Le applicazioni eseguite su un'istanza Amazon EC2 devono includere AWS credenziali nelle richieste API. AWS Potresti chiedere ai tuoi sviluppatori di archiviare AWS le credenziali direttamente all'interno dell'istanza Amazon EC2 e consentire alle applicazioni in quell'istanza di utilizzare tali credenziali. Tuttavia, in questo caso, gli sviluppatori dovrebbero gestire le credenziali, accertarsi

che vengano passate in modo sicuro a ciascuna istanza e aggiornare ogni istanza Amazon EC2 al momento di aggiornare le credenziali. Si tratta di una notevole quantità di lavoro aggiuntivo.

In alternativa, puoi (e devi) utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni eseguite in un'istanza Amazon EC2. Quando utilizzi un ruolo, non occorre distribuire credenziali a lungo termine (come, ad esempio, credenziali di accesso oppure chiavi di accesso) per un'istanza Amazon EC2. Al contrario, il ruolo fornisce autorizzazioni temporanee che le applicazioni possono utilizzare quando effettuano chiamate ad altre risorse. AWS Quando avvii un'istanza Amazon EC2, devi specificare un ruolo IAM da associare ad essa. Le applicazioni eseguite nell'istanza possono quindi utilizzare le credenziali provvisorie fornite dal ruolo per firmare le richieste API.

L'utilizzo dei ruoli per concedere autorizzazioni alle applicazioni eseguite nelle istanze Amazon EC2 richiede una configurazione leggermente più elaborata Un'applicazione in esecuzione su un'istanza Amazon EC2 viene astratta AWS dal sistema operativo virtualizzato. A causa di questa ulteriore separazione, è necessario un passaggio aggiuntivo per assegnare un AWS ruolo e le relative autorizzazioni a un'istanza Amazon EC2 e renderle disponibili alle sue applicazioni. Tale passaggio aggiuntivo prevede la creazione di un profilo dell'istanza collegato all'istanza. Il profilo dell'istanza contiene il ruolo e può fornire le credenziali provvisorie del ruolo a un'applicazione eseguita nell'istanza. Le credenziali provvisorie possono essere utilizzate nelle chiamate dell'API dell'applicazione per accedere alle risorse e limitare l'accesso alle sole risorse specificate dal ruolo.

> **Note**
>
> A un'istanza Amazon EC2 può essere assegnato soltanto un ruolo alla volta e tutte le applicazioni dell'istanza condividono lo stesso ruolo e le stesse autorizzazioni. Quando si utilizza Amazon ECS per gestire le istanze Amazon EC2, alle attività di Amazon ECS è possibile assegnare dei ruoli che possono essere distinti dal ruolo dell'istanza Amazon EC2 su cui è in esecuzione. L'assegnazione di un ruolo a ciascuna attività è conforme al principio dell'accesso con privilegi minimi e consente un controllo più granulare su operazioni e risorse. Per ulteriori informazioni, consulta la pagina Utilizzo dei ruoli IAM con le attività Amazon ECS nella Guida alle best practice per Amazon Elastic Container Service.

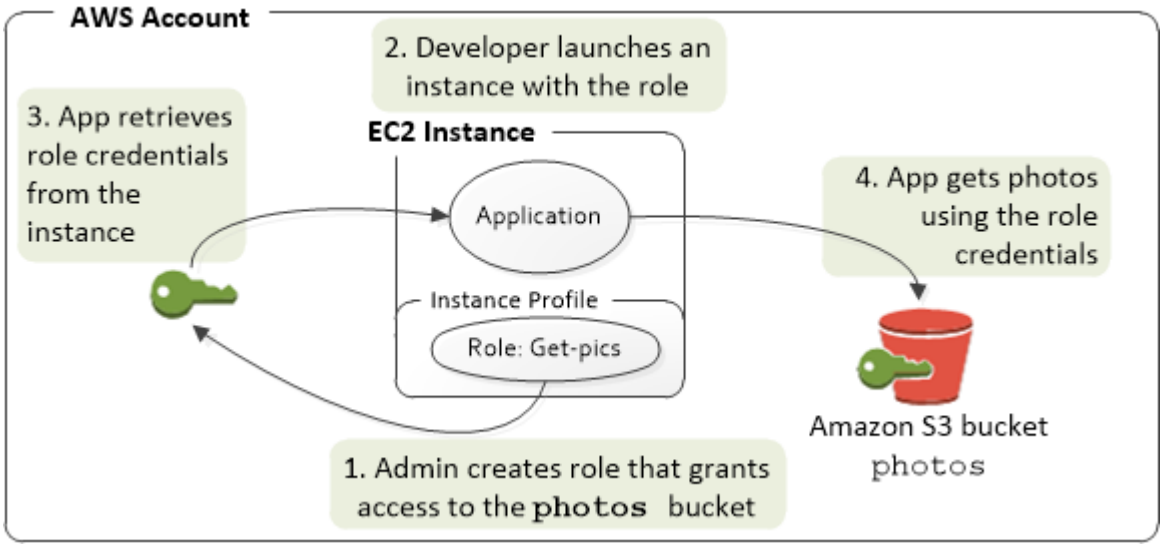
Questo tipo di utilizzo dei ruoli offre diversi vantaggi. Dato che le credenziali dei ruoli sono temporanee e vengono aggiornate automaticamente, non dovrai preoccuparti della gestione né dei rischi di sicurezza a lungo termine. Inoltre, se utilizzi un singolo ruolo per più istanze, quando apporti una modifica a un ruolo, queste si propaga automaticamente a tutte le istanze.

> **Note**
>
> Anche se in genere un ruolo viene assegnato a un'istanza Amazon EC2 all'avvio, puoi comunque effettuare il collegamento anche a un'istanza Amazon EC2 già in esecuzione. Per informazioni sul collegamento di un ruolo a un'istanza in esecuzione, consulta Ruoli IAM per Amazon EC2.

Argomenti

Funzionamento dei ruoli per le istanze Amazon EC2

Nella figura di seguito, uno sviluppatore esegue un'applicazione su un'istanza Amazon EC2 che richiede l'accesso al bucket S3 denominato `photos`. Un amministratore crea il ruolo di servizio `Get-pics` e lo collega all'istanza Amazon EC2. Il ruolo include una policy di autorizzazione che consente l'accesso in sola lettura al bucket S3 specificato. Include anche una policy di affidabilità che consente all'istanza Amazon EC2 di assumere il ruolo e recuperare le credenziali provvisorie. Quando l'applicazione viene eseguita sull'istanza, può utilizzare le credenziali provvisorie del ruolo per accedere al bucket delle foto. L'amministratore non ha bisogno di concedere allo sviluppatore l'autorizzazione di accedere al bucket delle foto e lo sviluppatore non si trova mai nella necessità di condividere o gestire credenziali.



1. L'amministratore utilizza IAM per creare il ruolo **`Get-pics`**. Nella policy di affidabilità del ruolo l'amministratore specifica che solo le istanze Amazon EC2 possono assumere quel ruolo. Nella policy di autorizzazione del ruolo l'amministratore specifica autorizzazioni di sola lettura per il bucket `photos`.
2. Uno sviluppatore avvia un'istanza Amazon EC2 e assegna il ruolo `Get-pics` all'istanza.

   > **Note**
   >
   > Se utilizzi la console IAM, il profilo dell'istanza viene gestito in automatico, con un processo quasi completamente trasparente. Tuttavia, se utilizzi l'API AWS CLI or per creare e gestire il ruolo e l'istanza Amazon EC2, devi creare il profilo dell'istanza e assegnargli il ruolo in fasi separate. Quindi, quando avvii l'istanza dovrai specificare il nome del profilo dell'istanza anziché il nome del ruolo.

3. Quando l'applicazione è in esecuzione, raccoglie le credenziali di sicurezza provvisorie dai metadati dell'istanza Amazon EC2, come descritto in Recupero delle credenziali di sicurezza dai metadati delle istanze. Si tratta di credenziali di sicurezza provvisorie che rappresentano il ruolo e hanno un periodo di validità limitato.

   Con alcuni AWS SDK, lo sviluppatore può utilizzare un provider per la gestione trasparente delle credenziali di sicurezza provvisorie. (La documentazione per i singoli AWS SDK descrive le funzionalità supportate da tale SDK per la gestione delle credenziali.)

   In alternativa, l'applicazione può ottenere le credenziali provvisorie direttamente dai metadati dell'istanza Amazon EC2. Le credenziali e i valori correlati sono disponibili nella categoria `iam/`

`security-credentials/`*`role-name`* (in questo caso `iam/security-credentials/Get-pics`) dei metadati. Se l'applicazione ottiene le credenziali dai metadati dell'istanza, può memorizzarle nella cache.

4. Grazie all'utilizzo delle credenziali provvisorie recuperate, l'applicazione può accedere al bucket delle foto. In virtù della policy collegata al ruolo **`Get-pics`** (Ottieni foto), l'applicazione dispone di autorizzazioni di sola lettura.

   Le credenziali di sicurezza temporanee disponibili nell'istanza vengono aggiornate automaticamente prima della scadenza, in modo da avere un set valido sempre disponibile. L'applicazione deve solo assicurarsi di ottenere un nuovo set di credenziali dai metadati dell'istanza prima della scadenza di quelle esistenti. È possibile utilizzare l' AWS SDK per gestire le credenziali in modo che l'applicazione non debba includere logica aggiuntiva per aggiornare le credenziali. Ad esempio, creando istanze di client con provider di credenziali del profilo dell'istanza. Tuttavia, se l'applicazione ottiene le credenziali di sicurezza provvisorie dai metadati dell'istanza e le memorizza nella cache, è necessario fornire un set di credenziali aggiornato ogni ora o almeno 15 minuti prima della scadenza del set corrente. L'ora di scadenza è indicata nelle informazioni restituite nella categoria `iam/security-credentials/`*`role-name`*.

Autorizzazioni richieste per l'utilizzo dei ruoli con Amazon EC2

Per avviare un'istanza con un ruolo, lo sviluppatore deve avere l'autorizzazione per avviare le istanze Amazon EC2 e per passare i ruoli IAM.

La seguente politica di esempio consente agli utenti di utilizzare per AWS Management Console avviare un'istanza con un ruolo. La policy include caratteri jolly (`*`) per consentire a un utente di passare qualsiasi ruolo ed eseguire tutte le operazioni di Amazon EC2 elencate. L'operazione `ListInstanceProfiles` consente agli utenti di visualizzare tutti i ruoli disponibili nell' Account AWS.

Example Esempio di policy che concede a un utente l'autorizzazione di utilizzare la console Amazon EC2 per avviare un'istanza con qualsiasi ruolo

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        },
        {
            "Sid": "ListEc2AndListInstanceProfiles",
            "Effect": "Allow",
            "Action": [
                "iam:ListInstanceProfiles",
                "ec2:Describe*",
                "ec2:Search*",
                "ec2:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitazione dei ruoli che possono essere passati alle istanze Amazon EC2 (utilizzando) PassRole

È possibile utilizzare l'autorizzazione `PassRole` per limitare i ruoli che un utente può passare a un'istanza Amazon EC2 quando avvia l'istanza. In questo modo si impedisce all'utente di eseguire le applicazioni che dispongono di più autorizzazioni rispetto a quelle concesse all'utente (ovvero di ottenere privilegi elevati). Ad esempio, immaginiamo che l'utente Alice disponga solo delle autorizzazioni per avviare le istanze Amazon EC2 e per operare con i bucket di Amazon S3, ma che passi a un'istanza Amazon EC2 un ruolo con autorizzazioni per operare con IAM e Amazon DynamoDB. In questo caso, Alice potrebbe essere in grado di avviare l'istanza, accedere a essa, ottenere credenziali di sicurezza temporanee e quindi eseguire operazioni IAM o DynamoDB per cui non dispone dell'autorizzazione.

Per limitare i ruoli che un utente può passare a un'istanza Amazon EC2, devi creare una policy che consenta l'operazione `PassRole`. A quel punto, puoi collegare la policy all'utente (o a un gruppo IAM a cui l'utente appartiene) che avvierà le istanze Amazon EC2. Nell'elemento `Resource` della policy devi elencare il ruolo o i ruoli che l'utente può passare alle istanze Amazon EC2. Quando l'utente avvia un'istanza e la associa a un ruolo, Amazon EC2 verifica se l'utente è autorizzato a inviare quel ruolo. Ovviamente, devi anche accertarti che il ruolo passato dall'utente non includa un numero di autorizzazioni maggiore di quello consentito all'utente.

> **Note**
>
> `PassRole` non è un'operazione API pari a `RunInstances` o `ListInstanceProfiles`. Si tratta invece di un'autorizzazione che AWS verifica ogni volta che l'ARN di un ruolo viene passato come parametro a un'API (o la console lo fa per conto dell'utente). In questo modo, un amministratore ha la possibilità di controllare quali ruoli possono essere passati dai vari utenti. In questo caso, garantisce che l'utente abbia l'autorizzazione per collegare un ruolo specifico a un'istanza Amazon EC2.

Example policy che concede a un utente l'autorizzazione di avviare un'istanza Amazon EC2 con un ruolo specifico

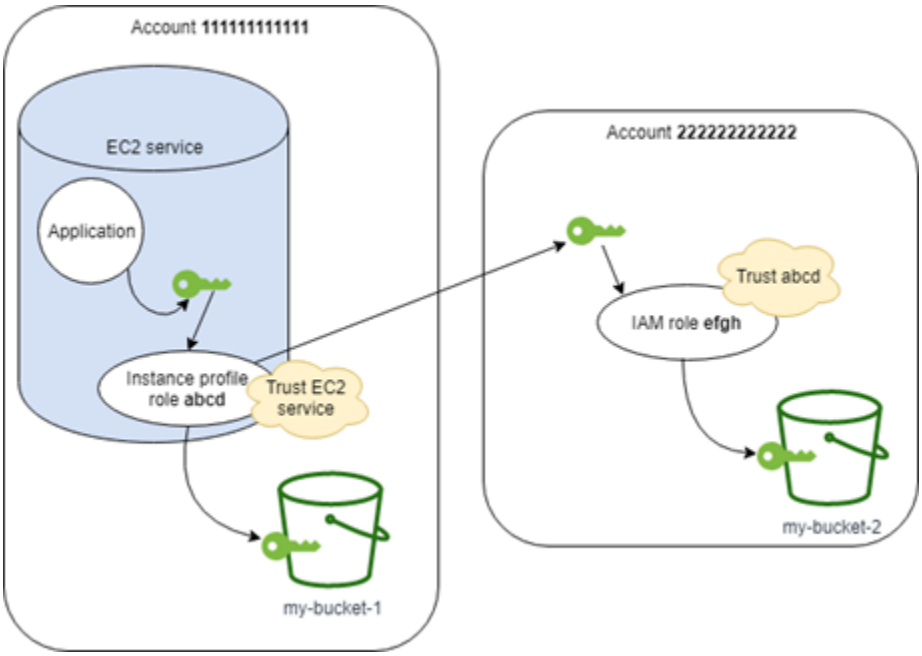
Il seguente esempio di policy consente agli utenti di utilizzare l'API Amazon EC2 per avviare un'istanza con un ruolo. L'elemento `Resource` specifica l'Amazon Resource Name (ARN) di un ruolo. Specificando l'ARN, la policy concede all'utente l'autorizzazione di passare solo il ruolo `Get-pics`. Se, all'avvio di un'istanza, l'utente cerca di specificare un ruolo diverso, l'operazione ha esito negativo. L'utente non è autorizzato a eseguire alcuna istanza, indipendentemente dal passaggio di un ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/Get-pics"
    }
  ]
}
```

Consentire a un ruolo del profilo dell'istanza di passare a un ruolo in un altro account

Puoi consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire comandi in un altro account. A tale scopo, devi consentire al ruolo dell'istanza Amazon EC2 nel primo account di passare a un ruolo nel secondo account.

Immagina di usarne due Account AWS e di voler consentire a un'applicazione in esecuzione su un'istanza Amazon EC2 di eseguire AWS CLIcomandi in entrambi gli account. Supponiamo che l'istanza Amazon EC2 esista nell'account `111111111111`. Tale istanza include il ruolo del profilo dell'istanza `abcd` che consente all'applicazione di eseguire attività Amazon S3 di sola lettura nel bucket `my-bucket-1` all'interno dello stesso account `111111111111`. Tuttavia, l'applicazione deve anche poter assumere il ruolo tra account `efgh` per accedere al bucket `my-bucket-2` di Amazon S3 nell'account `222222222222`.



Il ruolo del profilo dell'istanza Amazon EC2 `abcd` deve disporre della policy di autorizzazioni seguente per consentire all'applicazione di accedere al bucket `my-bucket-1` di Amazon S3:

Policy di autorizzazioni del ruolo ***abcd*** 111111111111 dell'account

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket-1/*",
                "arn:aws:s3:::my-bucket-1"
            ]
        },
        {
            "Sid": "AllowIPToAssumeCrossAccountRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::222222222222:role/efgh"
        }
    ]
}
```

Il ruolo `abcd` deve considerare il servizio Amazon EC2 come attendibile ad assumere il ruolo. A tale scopo, il ruolo `abcd` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***abcd*** dell'account 111111111111

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "abcdTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"Service": "ec2.amazonaws.com"}
        }
    ]
```

```
}
```

Supponiamo che il ruolo tra account `efgh` consenta attività Amazon S3 di sola lettura nel bucket `my-bucket-2` all'interno dello stesso account `222222222222`. A tale scopo, il ruolo tra account `efgh` deve disporre della seguente policy di autorizzazioni:

Policy di autorizzazioni del ruolo ***efgh*** 222222222222 dell'account

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccountLevelS3Actions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowListAndReadS3ActionOnMyBucket",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket-2/*",
                "arn:aws:s3:::my-bucket-2"
            ]
        }
    ]
}
```

Il ruolo `efgh` deve consentire al ruolo del profilo dell'istanza `abcd` di assumerlo. A tale scopo, il ruolo `efgh` deve disporre della seguente policy di attendibilità:

Policy di attendibilità del ruolo ***efgh*** dell'account 222222222222

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "efghTrustPolicy",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111111111111:role/abcd"}
        }
    ]
}
```

Come si inizia?

Per comprendere come funzionano i ruoli nelle istanze Amazon EC2, crea un ruolo con la console IAM, avvia un'istanza Amazon EC2 che usi tale ruolo e quindi osserva l'istanza in esecuzione. Puoi prendere in esame i metadati dell'istanza per consultare in che modo le credenziali provvisorie del ruolo vengano rese disponibili a un'istanza. Potrai anche consultare il modo in cui un'applicazione eseguita in un'istanza può utilizzare tale ruolo. Per ottenere ulteriori informazioni, usare le risorse indicate di seguito.

-
- Procedure guidate sugli SDK. La documentazione dell' AWS SDK include procedure dettagliate che mostrano un'applicazione in esecuzione su un'istanza Amazon EC2 che utilizza credenziali temporanee per i ruoli per leggere un bucket Amazon S3. Ogni procedura guidata presenta passaggi simili, ma utilizza un linguaggio di programmazione diverso:
  - Configurazione dei ruoli IAM per Amazon EC2 con SDK per Java nella Guida per gli sviluppatori di AWS SDK for Java
  - Avvio di un'istanza Amazon EC2 utilizzando SDK per .NET nella Guida per gli sviluppatori di AWS SDK for .NET
  - Creazione di una istanza Amazon EC2 con SDK for Ruby nella Guida per gli sviluppatori di AWS SDK for Ruby

Informazioni correlate

Per ulteriori informazioni sulla creazione di ruoli o di ruoli per le istanze Amazon EC2, consulta la seguente documentazione:

- Per ulteriori informazioni sull'utilizzo dei ruoli IAM con le istanze Amazon EC2, consulta la Amazon EC2 User Guide.

- Per creare un ruolo, consulta Creazione di ruoli IAM
- Per ulteriori informazioni sull'utilizzo delle credenziali di sicurezza provvisorie, vedi Credenziali di sicurezza temporanee in IAM.
- Se lavori con l'API IAM o la CLI, devi creare e gestire i profili delle istanze IAM. Per ulteriori informazioni sui profili delle istanze, consulta Utilizzo dei profili delle istanze.
- Per ulteriori informazioni sulle credenziali di sicurezza temporanee per i ruoli nei metadati dell'istanza, consulta Retrieving Security Credentials from Instance Metadata nella Amazon EC2 User Guide.

Utilizzo dei profili delle istanze

Utilizza un profilo dell'istanza per passare un ruolo IAM a un'istanza EC2. Per ulteriori informazioni, consulta i ruoli IAM per Amazon EC2 nella Amazon EC2 User Guide.

Gestione dei profili delle istanze (console)

Se utilizzi il per AWS Management Console creare un ruolo per Amazon EC2, la console crea automaticamente un profilo di istanza e gli assegna lo stesso nome del ruolo. Successivamente, quando utilizzi la console Amazon EC2 per avviare un'istanza con un ruolo IAM, potrai selezionare un ruolo da associare all'istanza. L'elenco visualizzato nella console è in effetti elenco di nomi di profili delle istanze. La console non crea un profilo dell'istanza per un ruolo non associato ad Amazon EC2.

Puoi utilizzare il AWS Management Console per eliminare i ruoli IAM e i profili di istanza per Amazon EC2 se il ruolo e il profilo dell'istanza hanno lo stesso nome. Per ulteriori informazioni sull'eliminazione dei profili di istanza, consulta Eliminazione di ruoli o profili delle istanze.

Gestione dei profili di istanza (AWS CLI o AWS API)

Se gestisci i tuoi ruoli dall' AWS CLI o dall' AWS API, crei ruoli e profili di istanza come azioni separate. Poiché ruoli e profili delle istanze possono avere nomi diversi, è importante che tu conosca i nomi dei profili e dei ruoli che contengono. In questo modo,sarà più semplice selezionare il corretto profilo all'avvio di un'istanza EC2.

È possibile collegare tag alle risorse IAM, inclusi i profili dell'istanza, per identificare, organizzare e controllare l'accesso a tali risorse. Puoi etichettare i profili delle istanze solo quando utilizzi l' AWS API AWS CLI o.

> **Note**
>
> Un profilo dell'istanza può contenere un solo ruolo IAM, mentre lo stesso ruolo può essere in più profili delle istanze. Non è possibile aumentare il numero di ruoli per profilo dell'istanza. Tuttavia, puoi rimuovere il ruolo esistente nel profilo dell'istanza e aggiungerne uno diverso. È quindi necessario attendere che la modifica appaia ovunque per AWS motivi di coerenza finale. Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Gestione dei profili delle istanze AWS CLI

È possibile utilizzare i seguenti AWS CLI comandi per lavorare con i profili di istanza in un AWS account.

- Per creare un profilo dell'istanza: `aws iam create-instance-profile`
- Applicare tag a un profilo dell'istanza: `aws iam tag-instance-profile`
- Elencare i tag per un profilo dell'istanza: `aws iam list-instance-profile-tags`
- Rimuovere i tag da un profilo dell'istanza: `aws iam untag-instance-profile`
- Per aggiungere un ruolo a un profilo dell'istanza: `aws iam add-role-to-instance-profile`
- Per elencare i profili delle istanze: `aws iam list-instance-profiles`, `aws iam list-instance-profiles-for-role`
- Per ottenere informazioni su un profilo dell'istanza: `aws iam get-instance-profile`
- Per rimuovere un ruolo da un profilo dell'istanza: `aws iam remove-role-from-instance-profile`
- Per eliminare un profilo dell'istanza: `aws iam delete-instance-profile`

È anche possibile collegare un ruolo a un'istanza EC2 già in esecuzione, utilizzando i comandi riportati di seguito. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2.

- Collegare un profilo dell'istanza con ruolo a un'istanza EC2 in funzione o arrestata: `aws ec2 associate-iam-instance-profile`
- Ottenere informazioni su un profilo dell'istanza collegato a un'istanza EC2: `aws ec2 describe-iam-instance-profile-associations`

- Distaccare un profilo dell'istanza con ruolo da un'istanza EC2 in funzione o arrestata: `aws ec2 disassociate-iam-instance-profile`

Gestione dei profili delle istanze (API AWS)

Puoi chiamare le seguenti operazioni AWS API per lavorare con i profili di istanza in un Account AWS.

- Per creare un profilo dell'istanza: `CreateInstanceProfile`
- Applicare tag a un profilo dell'istanza: `TagInstanceProfile`
- Elencare i tag su un profilo dell'istanza: `ListInstanceProfileTags`
- Rimuovere i tag da un profilo dell'istanza: `UntagInstanceProfile`
- Per aggiungere un ruolo a un profilo dell'istanza: `AddRoleToInstanceProfile`
- Per elencare i profili delle istanze: `ListInstanceProfiles`, `ListInstanceProfilesForRole`
- Per ottenere informazioni su un profilo dell'istanza: `GetInstanceProfile`
- Per rimuovere un ruolo da un profilo dell'istanza: `RemoveRoleFromInstanceProfile`
- Per eliminare un profilo dell'istanza: `DeleteInstanceProfile`

È anche possibile collegare un ruolo a un'istanza EC2 già in esecuzione, richiamando le operazioni riportate di seguito. Per ulteriori informazioni, consulta Ruoli IAM per Amazon EC2.

- Collegare un profilo dell'istanza con ruolo a un'istanza EC2 in funzione o arrestata: `AssociateIamInstanceProfile`
- Ottenere informazioni su un profilo dell'istanza collegato a un'istanza EC2: `DescribeIamInstanceProfileAssociations`
- Distaccare un profilo dell'istanza con ruolo da un'istanza EC2 in funzione o arrestata: `DisassociateIamInstanceProfile`

Revoca delle credenziali di sicurezza temporanee per i ruoli IAM

> **Warning**
>
> Se segui i passaggi in questa pagina, a tutti gli utenti con sessioni correnti create assumendo il ruolo viene negato l'accesso a tutte le AWS azioni e le risorse. Questo può causare la perdita di dati non salvati da parte degli utenti.

Quando consenti agli utenti di accedere a sessioni AWS Management Console con una durata di sessione lunga (ad esempio 12 ore), le loro credenziali temporanee non scadono così rapidamente. Se gli utenti espongono inavvertitamente le proprie credenziali a una terza parte non autorizzata, tale parte ha accesso per la durata della sessione. Tuttavia, è possibile revocare immediatamente tutte le autorizzazioni per le credenziali del ruolo rilasciate prima di un certo periodo di tempo, se necessario. Tutte le credenziali temporanee per quel ruolo emesse prima del momento specificata diventano non valide. Questo costringe tutti gli utenti a ripetere l'autenticazione e a richiedere nuove credenziali.

> **Note**
>
> Non è possibile revocare la sessione per un ruolo collegato ai servizi.

Quando si revocano le autorizzazioni per un ruolo utilizzando la procedura AWS riportata in questo argomento, al ruolo viene associata una nuova politica in linea che nega tutte le autorizzazioni a tutte le azioni. Include una condizione che applica le restrizioni solo se l'utente ha assunto il ruolo prima del momento in cui sono state revocate le autorizzazioni. Se l'utente assume il ruolo dopo la revoca delle autorizzazioni, la policy di rifiuto non si applica a quell'utente.

Per ulteriori informazioni sulla negazione dell’accesso, consulta Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee.

> **Important**
>
> La policy di rifiuto si applica a tutti gli utenti con il ruolo specificato, non solo a quelli con sessioni della console di durata più lunga.

Autorizzazioni minime per revocare le autorizzazioni di sessione da un ruolo

Per revocare le autorizzazioni di sessione da un ruolo, è necessario disporre dell'autorizzazione `PutRolePolicy` per il ruolo. In questo modo è possibile collegare la policy inline `AWSRevokeOlderSessions` al ruolo.

Revoca delle autorizzazioni di sessione

È possibile revocare le autorizzazioni di sessione da un ruolo per negare tutte le autorizzazioni a qualsiasi utente che ha assunto il ruolo.

> **Note**
>
> Non è possibile modificare i ruoli in IAM creati dai set di autorizzazioni di IAM Identity Center. È necessario revocare la sessione attiva del set di autorizzazioni per un utente in IAM Identity Center. Per ulteriori informazioni, consulta Revoca le sessioni di ruolo IAM attive create dai set di autorizzazioni nella Guida per l'utente di IAM Identity Center.

Per rifiutare immediatamente tutte le autorizzazioni a qualsiasi utente corrente con credenziali del ruolo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione scegli Ruoli, quindi seleziona il nome (non la casella di controllo) del ruolo per cui desideri revocare le autorizzazioni.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, selezionare la scheda Revoke sessions (Revoca sessioni).
4. Nelle scheda Revoke sessions (Revoca sessioni) selezionare Revoke active sessions (Revoca sessioni attive).
5. AWS ti chiede di confermare l'azione. Seleziona la casella di controllo I acknowledge that I am revoking all active sessions for this role. (Riconosco che sto revocando tutte le sessioni attive per questo ruolo.) e scegli Revoke active sessions (Revoca le sessioni attive) nella finestra di dialogo.

   IAM quindi allega una policy denominata `AWSRevokeOlderSessions` al ruolo. Dopo aver scelto Revoca sessioni attive, la policy nega l'accesso agli utenti che hanno assunto il ruolo in

passato e per circa 30 secondi nel futuro. Questa scelta temporale futura tiene conto del ritardo di propagazione della politica per gestire una nuova sessione acquisita o rinnovata prima che la politica aggiornata entrasse in vigore in una determinata regione. Qualsiasi utente che assume il ruolo più di circa 30 secondi dopo aver scelto Revoca sessioni attive non ne risente. Per scoprire perché le modifiche non sono sempre immediatamente visibili, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

> **Note**
>
> Se scegli di revocare nuovamente le sessioni attive in un secondo momento, la data e l'ora nella politica vengono aggiornate e vengono nuovamente negate tutte le autorizzazioni a tutti gli utenti che hanno assunto il ruolo prima del nuovo orario specificato.

Gli utenti validi le cui sessioni sono revocate in questo modo devono acquisire credenziali provvisorie per una nuova sessione per continuare a lavorare. La AWS CLI memorizza nella cache le credenziali finché non scadono. Per forzare la CLI a eliminare e aggiornare le credenziali memorizzate nella cache che non sono più valide, eseguire uno dei seguenti comandi:

Linux, macOS o Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

Revoca delle autorizzazioni di sessione prima di un orario specificato

Puoi anche revocare le autorizzazioni di sessione in qualsiasi momento a tua scelta utilizzando l'SDK AWS CLI o l'SDK per specificare un valore per la chiave nell'elemento Condition di una policy. `aws:TokenIssue Ora`

Questa policy nega tutte le autorizzazioni, quando il valore di `aws:TokenIssueTime` è precedente alla data e ora specificate. Il valore di `aws:TokenIssueTime` corrisponde al momento in cui sono state create le credenziali di sicurezza provvisorie. Il `aws:TokenIssueTime` valore è presente solo nel contesto delle AWS richieste firmate con credenziali di sicurezza temporanee, pertanto l'istruzione

Deny nella policy non influisce sulle richieste firmate con le credenziali a lungo termine dell'utente IAM.

Questa policy può essere collegata a un ruolo. In questo caso, la policy influisce solo sulle credenziali di sicurezza provvisorie create da tale ruolo prima della data e ora specificate.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {"aws:TokenIssueTime": "2014-05-07T23:47:00Z"}
    }
  }
}
```

Gli utenti validi le cui sessioni sono revocate in questo modo devono acquisire credenziali provvisorie per una nuova sessione per continuare a lavorare. Le credenziali vengono memorizzate AWS CLI nella cache fino alla loro scadenza. Per forzare la CLI a eliminare e aggiornare le credenziali memorizzate nella cache che non sono più valide, eseguire uno dei seguenti comandi:

Linux, macOS o Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

Gestione di ruoli IAM

Occasionalmente è necessario modificare o eliminare i ruoli creati. Per modificare un ruolo, è possibile eseguire una delle operazioni seguenti:

- Modificare le policy associate al ruolo
- Modificare gli utenti che possono accedere al ruolo
- Modificare le autorizzazioni che il ruolo concede agli utenti

- Modifica l'impostazione della durata massima della sessione per i ruoli assunti utilizzando l'API AWS Management Console, AWS CLI o

È anche possibile eliminare i ruoli che non sono più necessari. Puoi gestire i tuoi ruoli dalla AWS Management Console AWS CLI, dalla e dall'API.

Argomenti

- Modifica di un ruolo
- Eliminazione di ruoli o profili delle istanze

Modifica di un ruolo

Puoi utilizzare l'API AWS Management Console AWS CLI, the o IAM per apportare modifiche a un ruolo.

Argomenti

- Visualizzazione dell'accesso per il ruolo
- Generazione di una policy basata sulle informazioni di accesso
- Modifica di un ruolo (console)
- Modifica di un ruolo (AWS CLI)
- Modifica di un ruolo (AWS API)

Visualizzazione dell'accesso per il ruolo

Prima di modificare le autorizzazioni per un ruolo, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Generazione di una policy basata sulle informazioni di accesso

Talvolta, è possibile concedere autorizzazioni a un'entità IAM (utente o ruolo) oltre a quelle richieste. Per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile

utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy basate sull'attività di accesso.

Modifica di un ruolo (console)

È possibile utilizzare il AWS Management Console per modificare un ruolo. Per modificare il set di tag su un ruolo, consulta Gestione dei tag sui ruoli IAM (console).

Argomenti

- Modifica di una policy di attendibilità del ruolo (Console)
- Modifica di una policy di autorizzazioni del ruolo (Console)
- Modifica di una descrizione del ruolo (Console)
- Modifica della durata massima della sessione di un ruolo (Console)
- Modifica di un limite delle autorizzazioni di un ruolo (Console)

Modifica di una policy di attendibilità del ruolo (Console)

Per cambiare l'utente che può assumere un ruolo, modifica la policy di affidabilità del ruolo. Non puoi modificare la policy di attendibilità per un ruolo collegato al servizio.

> **Note**
>
> - Se un utente viene elencato come principale in una policy di attendibilità del ruolo ma non può assumere il ruolo, controlla il limite delle autorizzazioni dell'utente. Se è impostato un limite delle autorizzazioni per l'utente, questo deve consentire l'operazione `sts:AssumeRole`.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e aggiornano automaticamente tali credenziali. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale.

Per modificare una policy di attendibilità del ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Ruoli.
3. Nell'elenco di ruoli dell'account selezionare il nome del ruolo da modificare.
4. Scegli la scheda Relazioni di attendibilità e quindi Modifica policy di attendibilità.
5. Modificare la policy di affidabilità in base alle esigenze. Per aggiungere ulteriori entità principali che possono assumere il ruolo, specificarle nell'elemento `Principal`. Ad esempio, il seguente frammento di policy mostra come fare riferimento a due Account AWS nell'`Principal`elemento:

   ```
   "Principal": {
     "AWS": [
       "arn:aws:iam::111122223333:root",
       "arn:aws:iam::444455556666:root"
     ]
   },
   ```

   Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concessione di autorizzazioni agli utenti per il cambio di ruoli.

   Il seguente frammento di policy mostra come fare riferimento a due AWS servizi nell'elemento: `Principal`

   ```
   "Principal": {
     "Service": [
       "opsworks.amazonaws.com",
       "ec2.amazonaws.com"
     ]
   ```

```
},
```

6. Una volta completata la modifica della policy di attendibilità, scegli Update policy (Aggiorna policy) per salvare le modifiche.

   Per ulteriori informazioni sulla sintassi e sulla struttura della policy, consultare Policy e autorizzazioni in IAM e Documentazione di riferimento degli elementi delle policy JSON IAM.

Per permettere agli utenti in un account esterno attendibile di usare il ruolo (console)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

1. Accedi a un sito esterno affidabile. Account AWS
2. Stabilire se collegare le autorizzazioni a un utente o a un gruppo. Nel riquadro di navigazione della console IAM, scegli Users (Utenti) o User groups (Gruppi di utenti) in base alle esigenze.
3. Scegliere il nome dell'utente o del gruppo a cui si desidera concedere l'accesso e selezionare la scheda Permissions (Autorizzazioni).
4. Esegui una di queste operazioni:

   - Per modificare una policy gestita dal cliente, selezionare il nome della policy, Edit policy (Modifica policy) e la scheda JSON. Non è possibile modificare una politica AWS gestita. AWS le politiche gestite vengono visualizzate con l' AWS icona (). Per ulteriori informazioni sulla differenza tra politiche AWS gestite e politiche gestite dal cliente, vedereePolicy gestite e policy inline.
   - Per modificare una policy inline, selezionare la freccia accanto al nome della policy e scegliere Edit policy (Modifica policy).
5. Nell'editor di policy aggiungere un nuovo elemento `Statement` che specifica quanto segue:

```
{
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": "arn:aws:iam::ACCOUNT-ID:role/ROLE-NAME"
}
```

   Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

6. Seguire le indicazioni sullo schermo per completare la modifica della policy.

Modifica di una policy di autorizzazioni del ruolo (Console)

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi che funzionano con IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Per modificare le autorizzazioni permesse da un ruolo (console)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Ruoli.
3. Selezionare il nome del ruolo da modificare e la scheda Permissions (Autorizzazioni).
4. Esegui una di queste operazioni:

   - Per modificare una policy gestita dal cliente esistente, selezionare il nome della policy e scegliere Edit policy (Modifica policy).

     > **Note**
     >
     > Non è possibile modificare una politica AWS gestita. AWS le politiche gestite vengono visualizzate con l' AWS icona ().
     >
     > Per ulteriori informazioni sulle differenze tra le policy gestite da AWS e quelle gestite dal cliente, consultare Policy gestite e policy inline.

   - Per collegare una policy gestita esistente al ruolo, scegli Add permissions (Aggiungi autorizzazioni) e quindi Attach policies (Collega policy).
   - Per modificare una policy inline esistente, espandi la policy e scegli Edit (Modifica).
   - Per integrare una nuova policy inline, scegli Add permissions (Aggiungi autorizzazioni), quindi Create inline policy (Crea policy inline).
   - Per rimuovere una politica esistente dal ruolo, seleziona la casella di controllo accanto al nome della politica, quindi scegli Rimuovi.

Modifica di una descrizione del ruolo (Console)

Per cambiare la descrizione del ruolo, modifica il testo di descrizione.

Per modificare la descrizione di un ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
3. Scegliere il nome del ruolo da modificare.
4. Nella sezione Summary (Riepilogo), scegli Edit (Modifica).
5. Digita una nuova descrizione nella casella e scegli Save changes (Salva modifiche).

Modifica della durata massima della sessione di un ruolo (Console)

Per specificare l'impostazione della durata massima della sessione per i ruoli che vengono assunti utilizzando la console AWS CLI, l'o l' AWS API, modifica il valore di impostazione della durata massima della sessione. Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Se non specifichi un valore, viene applicata l'impostazione predefinita massima di 1 ora. Questa impostazione non limita le sessioni assunte dai servizi AWS .

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti utilizzando la console o l' AWS API (console) AWS CLI

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
3. Scegliere il nome del ruolo da modificare.
4. Nella sezione Summary (Riepilogo), scegli Edit (Modifica).
5. In Maximum session duration (Durata massima della sessione), scegli un valore. In alternativa, scegli Custom duration (Durata personalizzata) e inserisci un valore (in secondi).
6. Seleziona Salvataggio delle modifiche.

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

Per impostazione predefinita AWS Management Console, le sessioni utente IAM durano 12 ore. Agli utenti IAM viene che cambiano ruoli nella console viene concessa la durata massima della sessione del ruolo o il tempo rimanente nella sessione dell'utente IAM, a seconda di quale sia minore.

Chiunque assuma il ruolo dell' AWS API AWS CLI or può richiedere una sessione più lunga, fino a questo massimo. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta.

- Per specificare la durata di una sessione AWS CLI utilizzando il `duration-seconds` parametro. Per ulteriori informazioni, consulta Passaggio a un ruolo IAM (AWS CLI).
- Per specificare la durata di una sessione utilizzando l' AWS API, utilizza il `DurationSeconds` parametro. Per ulteriori informazioni, consulta Passaggio a un ruolo IAM (AWS API).

Modifica di un limite delle autorizzazioni di un ruolo (Console)

Per modificare il numero massimo di autorizzazioni consentite per un ruolo, modifica il limite delle autorizzazioni del ruolo.

Per modificare la policy utilizzata per impostare il limite delle autorizzazioni per un ruolo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Scegli il nome del ruolo con il limite delle autorizzazioni che desideri modificare.
4. Scegli la scheda Autorizzazioni. Se necessario, aprire la sezione Permissions boundary (Limite delle autorizzazioni) e selezionare Change boundary (Modifica limite).
5. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
6. Selezionare Change boundary (Modifica limite).

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo.

Modifica di un ruolo (AWS CLI)

È possibile utilizzare il AWS Command Line Interface per modificare un ruolo. Per modificare il set di tag su un ruolo, consulta Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM.

Argomenti

- Modifica di una policy di attendibilità del ruolo (AWS CLI)
- Modifica di una policy di autorizzazioni del ruolo (AWS CLI)
- Modifica di una descrizione del ruolo (AWS CLI)
- Modifica della durata massima della sessione di un ruolo (AWS CLI)
- Modifica di un limite delle autorizzazioni di un ruolo (AWS CLI)

Modifica di una policy di attendibilità del ruolo (AWS CLI)

Per cambiare l'utente che può assumere un ruolo, modifica la policy di affidabilità del ruolo. Non puoi modificare la policy di attendibilità per un ruolo collegato al servizio.

> **Note**
>
> - Se un utente viene elencato come principale in una policy di attendibilità del ruolo ma non può assumere il ruolo, controlla il limite delle autorizzazioni dell'utente. Se è impostato un limite delle autorizzazioni per l'utente, questo deve consentire l'operazione `sts:AssumeRole`.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e aggiornano automaticamente tali credenziali. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di attendibilità dei ruoli per aggiungere il ruolo principale ARN o Account AWS ARN, vedere. Modifica di una policy di attendibilità del ruolo (Console)

Per modificare una policy di attendibilità del ruolo (AWS CLI)

1. (Facoltativo) Se non si conosce il nome del ruolo da modificare, eseguire il comando seguente per elencare i ruoli nell'account:
   - aws iam list-roles

2. (Facoltativo) Per visualizzare la policy di affidabilità corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role

3. Per modificare le entità principali attendibili che possono accedere al ruolo, creare un file di testo con la policy di affidabilità aggiornata. È possibile usare qualsiasi editor di testo per creare la policy.

   Ad esempio, la seguente politica di fiducia mostra come fare riferimento a due Account AWS nell'`Principal`elemento. In questo modo gli utenti all'interno di due diversi Account AWS possono assumere questo ruolo.

   ```
   {
       "Version": "2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Principal": {"AWS": [
               "arn:aws:iam::111122223333:root",
               "arn:aws:iam::444455556666:root"
           ]},
           "Action": "sts:AssumeRole"
       }
   }
   ```

   Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concessione di autorizzazioni agli utenti per il cambio di ruoli.

4. Per utilizzare il file creato per aggiornare la policy di attendibilità, eseguire il comando seguente:

   - era io update-assume-role-policy

Per permettere agli utenti in un account esterno attendibile di usare il ruolo (AWS CLI)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

1. Creare un file JSON contenente una policy di autorizzazione che concede le autorizzazioni ad assumere il ruolo. La policy seguente contiene ad esempio le autorizzazioni minime necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::ACCOUNT-ID-THAT-CONTAINS-ROLE:role/ROLE-NAME"
  }
}
```

   Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

2. Esegui il comando seguente per caricare il file JSON contenente la policy di attendibilità in IAM:

   - aws iam create-policy

   L'output di questo comando include l'ARN della policy. Prendere nota di questo ARN, perché sarà necessario in una fase successiva.

3. Stabilire a quale utente o gruppo collegare la policy. Se non si conosce il nome dell'utente o del gruppo desiderato, usare uno dei comandi seguenti per elencare gli utenti o i gruppi nell'account:

   - aws iam list-users
   - aws iam list-groups

4. Usare uno dei comandi seguenti per collegare la policy creata nel passaggio precedente all'utente o al gruppo:

   - era io attach-user-policy
   - era io attach-group-policy

Modifica di una policy di autorizzazioni del ruolo (AWS CLI)

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi che funzionano con IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Per modificare le autorizzazioni permesse da un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare le autorizzazioni correnti associate a un ruolo, eseguire i comandi seguenti:
    1. aws iam list-role-policies per elencare le politiche in linea
    2. aws mira list-attached-role-policies a elencare le politiche gestite
2. Il comando per aggiornare le autorizzazioni per il ruolo varia a seconda del fatto che si aggiorni una policy gestita o una policy inline.

    Per aggiornare una policy gestita, eseguire il comando seguente per creare una nuova versione della policy gestita:

    - era io create-policy-version

    Per aggiornare una policy inline, eseguire il comando seguente:

    - era io put-role-policy

Modifica di una descrizione del ruolo (AWS CLI)

Per cambiare la descrizione del ruolo, modifica il testo di descrizione.

Per modificare la descrizione di un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, eseguire il comando seguente:

    - aws iam get-role

2. Per aggiornare la descrizione di un ruolo, eseguire il comando seguente con il parametro relativo alla descrizione:

   - aws iam update-role

Modifica della durata massima della sessione di un ruolo (AWS CLI)

Per specificare l'impostazione della durata massima della sessione per i ruoli assunti tramite AWS CLI o API, modifica il valore di tale impostazione. Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Se non specifichi un valore, viene applicata l'impostazione predefinita massima di 1 ora. Questa impostazione non limita le sessioni assunte dai AWS servizi.

> **Note**
>
> Chiunque assuma il ruolo dell'API AWS CLI o può utilizzare il parametro `duration-seconds` CLI o `DurationSeconds` il parametro API per richiedere una sessione più lunga. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta usando il parametro `DurationSeconds`. Se gli utenti non specificano un valore per il parametro `DurationSeconds` le loro credenziali di sicurezza rimangono valide per un'ora.

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti tramite AWS CLI (AWS CLI)

1. (Facoltativo) Per visualizzare l'impostazione della durata massima della sessione corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role

2. Per aggiornare l'impostazione della durata massima della sessione di un ruolo, eseguire il comando seguente con il parametro `max-session-duration` della CLI oppure il parametro API `MaxSessionDuration`:

   - aws iam update-role

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

Modifica di un limite delle autorizzazioni di un ruolo (AWS CLI)

Per modificare il numero massimo di autorizzazioni consentite per un ruolo, modifica il limite delle autorizzazioni del ruolo.

Per modificare la policy gestita utilizzata per impostare il limite delle autorizzazioni per un ruolo (AWS CLI)

1. (Facoltativo) Per visualizzare il limite delle autorizzazioni corrente per un ruolo, eseguire il comando seguente:

   - aws iam get-role
2. Per usare un'altra policy gestita per aggiornare il limite delle autorizzazioni per un ruolo, eseguire il comando seguente:

   - era aim put-role-permissions-boundary

   Un ruolo può avere solo una policy gestita impostata come limite delle autorizzazioni. Modificando il limite delle autorizzazioni è possibile modificare il numero massimo di autorizzazioni consentite per un ruolo.

Modifica di un ruolo (AWS API)

È possibile utilizzare l' AWS API per modificare un ruolo. Per modificare il set di tag su un ruolo, consulta Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM.

Argomenti

- Modifica di una politica di fiducia (AWS API) per i ruoli
- Modifica di una politica di autorizzazione dei ruoli (API)AWS
- Modifica di una descrizione del ruolo (API AWS)
- Modifica della durata massima della sessione (AWS API) di un ruolo
- Modifica del limite delle autorizzazioni di ruolo (API)AWS

Modifica di una politica di fiducia (AWS API) per i ruoli

Per cambiare l'utente che può assumere un ruolo, modifica la policy di affidabilità del ruolo. Non puoi modificare la policy di attendibilità per un ruolo collegato al servizio.

> **Note**
>
> - Se un utente viene elencato come principale in una policy di attendibilità del ruolo ma non può assumere il ruolo, controlla il limite delle autorizzazioni dell'utente. Se è impostato un limite delle autorizzazioni per l'utente, questo deve consentire l'operazione `sts:AssumeRole`.
> - Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e aggiornano automaticamente tali credenziali. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di attendibilità dei ruoli per aggiungere il ruolo principale ARN o Account AWS ARN, vedere. Modifica di una policy di attendibilità del ruolo (Console)

Per modificare una politica di attendibilità dei ruoli (API)AWS

1. (Facoltativo) Se non si conosce il nome del ruolo che si desidera modificare, chiamare l'operazione seguente per elencare i ruoli nell'account:
   - ListRoles
2. (Facoltativo) Per visualizzare la policy di affidabilità corrente per un ruolo, chiamare l'operazione seguente:
   - GetRole
3. Per modificare le entità principali attendibili che possono accedere al ruolo, creare un file di testo con la policy di affidabilità aggiornata. È possibile usare qualsiasi editor di testo per creare la policy.

   Ad esempio, la seguente politica di fiducia mostra come fare riferimento a due Account AWS nell'`Principal`elemento. In questo modo gli utenti all'interno di due diversi Account AWS possono assumere questo ruolo.

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"AWS": [
            "arn:aws:iam::111122223333:root",
            "arn:aws:iam::444455556666:root"
        ]},
        "Action": "sts:AssumeRole"
    }
}
```

Se specifichi un'entità principale in un altro account, l'aggiunta di un account alla policy di attendibilità di un ruolo è solo una parte della creazione della relazione di trust tra più account. Per impostazione predefinita, gli utenti negli account attendibili non possono assumere il ruolo. L'amministratore del nuovo account attendibile deve concedere agli utenti l'autorizzazione ad assumere il ruolo. A tale scopo, l'amministratore deve creare o modificare una policy collegata all'utente per consentire all'utente di accedere all'operazione `sts:AssumeRole`. Per ulteriori informazioni, consultare la procedura seguente o Concessione di autorizzazioni agli utenti per il cambio di ruoli.

4. Per utilizzare il file creato per aggiornare la policy di attendibilità, chiamare l'operazione seguente:

   - UpdateAssumeRolePolicy

Per consentire agli utenti di un account esterno affidabile di utilizzare il ruolo (AWS API)

Per ulteriori informazioni e dettagli su questa procedura, consultare Concessione di autorizzazioni agli utenti per il cambio di ruoli.

1. Creare un file JSON contenente una policy di autorizzazione che concede le autorizzazioni ad assumere il ruolo. La policy seguente contiene ad esempio le autorizzazioni minime necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::ACCOUNT-ID-THAT-CONTAINS-ROLE:role/ROLE-NAME"
  }
```

```
}
```

Sostituire l'ARN nell'istruzione con l'ARN del ruolo che l'utente può assumere.

2. Chiama l'operazione seguente per caricare il file JSON contenente la policy di attendibilità in IAM:

   - CreatePolicy

   L'output di questa operazione include l'ARN della policy. Prendere nota di questo ARN, perché sarà necessario in una fase successiva.

3. Stabilire a quale utente o gruppo collegare la policy. Se non si conosce il nome dell'utente o del gruppo desiderato, chiamare una delle operazioni seguenti per elencare gli utenti o i gruppi nell'account:

   - ListUsers
   - ListGroups

4. Chiamare una delle operazioni seguenti per collegare la policy creata nel passaggio precedente all'utente o al gruppo:

   - API: AttachUserPolicy
   - AttachGroupPolicy

Modifica di una politica di autorizzazione dei ruoli (API)AWS

Per modificare le autorizzazioni permesse dal ruolo, modifica la policy (o le policy) di autorizzazioni del ruolo. Non è possibile modificare la policy di autorizzazione per un ruolo collegato ai servizi in IAM. Potresti essere in grado di modificare la policy di autorizzazione all'interno del servizio che dipende dal ruolo. Per controllare se un servizio supporta questa funzionalità, consulta AWS servizi che funzionano con IAM e individua i servizi che hanno Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Per modificare le autorizzazioni consentite da un ruolo (API)AWS

1. (Facoltativo) Per visualizzare le autorizzazioni correnti associate a un ruolo, chiamare le operazioni seguenti:

1. ListRolePoliciesper elencare le politiche in linea
2. ListAttachedRolePoliciesper elencare le politiche gestite

2. L'operazione per aggiornare le autorizzazioni per il ruolo varia a seconda del fatto che si aggiorni una policy gestita o una policy inline.

   Per aggiornare una policy gestita, chiamare l'operazione seguente per creare una nuova versione della policy gestita:

   - CreatePolicyVersion

   Per aggiornare una policy inline, chiamare l'operazione seguente:

   - PutRolePolicy

Modifica di una descrizione del ruolo (API AWS)

Per cambiare la descrizione del ruolo, modifica il testo di descrizione.

Per modificare la descrizione di un ruolo (AWS API)

1. (Facoltativo) Per visualizzare la descrizione corrente per un ruolo, chiamare l'operazione seguente:

   - GetRole

2. Per aggiornare la descrizione di un ruolo, chiamare l'operazione seguente con il parametro relativo alla descrizione:

   - UpdateRole

Modifica della durata massima della sessione (AWS API) di un ruolo

Per specificare l'impostazione della durata massima della sessione per i ruoli assunti tramite AWS CLI o API, modifica il valore di tale impostazione. Questa impostazione può avere un valore compreso tra 1 ora e 12 ore. Se non specifichi un valore, viene applicata l'impostazione predefinita massima di 1 ora. Questa impostazione non limita le sessioni assunte dai AWS servizi.

> **Note**
>
> Chiunque assuma il ruolo dell'API AWS CLI o può utilizzare il parametro `duration-seconds` CLI o `DurationSeconds` il parametro API per richiedere una sessione più lunga. L'impostazione `MaxSessionDuration` determina la durata massima della sessione del ruolo che può essere richiesta usando il parametro `DurationSeconds`. Se gli utenti non specificano un valore per il parametro `DurationSeconds` le loro credenziali di sicurezza rimangono valide per un'ora.

Per modificare l'impostazione della durata massima della sessione per i ruoli assunti utilizzando l'API (API)AWS

1. (Facoltativo) Per visualizzare l'impostazione della durata massima della sessione corrente per un ruolo, chiamare l'operazione seguente:

   - GetRole
2. Per aggiornare l'impostazione della durata massima della sessione di un ruolo, chiamare l'operazione seguente con il parametro `max-sessionduration` della CLI oppure il parametro API `MaxSessionDuration`:

   - UpdateRole

   Le modifiche non verranno applicate fino alla volta successiva in cui qualcuno assume questo ruolo. Per informazioni su come revocare le sessioni esistenti per il ruolo, consultare Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

Modifica del limite delle autorizzazioni di ruolo (API)AWS

Per modificare il numero massimo di autorizzazioni consentite per un ruolo, modifica il limite delle autorizzazioni del ruolo.

Per modificare la policy gestita utilizzata per impostare il limite delle autorizzazioni per un ruolo (AWS API)

1. (Facoltativo) Per visualizzare il limite delle autorizzazioni corrente per un ruolo, richiamare l'operazione seguente:

- GetRole

2. Per usare un'altra policy gestita per aggiornare il limite delle autorizzazioni per un ruolo, chiamare l'operazione seguente:

   - PutRolePermissionsBoundary

   Un ruolo può avere solo una policy gestita impostata come limite delle autorizzazioni. Modificando il limite delle autorizzazioni è possibile modificare il numero massimo di autorizzazioni consentite per un ruolo.

Eliminazione di ruoli o profili delle istanze

Se un ruolo non è più necessario, si consiglia di eliminare il ruolo e le autorizzazioni associate. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente.

Se il ruolo è stato associato a un'istanza EC2, è anche possibile rimuovere il ruolo dal profilo dell'istanza e quindi eliminare il profilo dell'istanza.

> **Warning**
>
> Assicurati di non avere istanze Amazon EC2 in esecuzione con il ruolo o il profilo di istanza che stai per eliminare. L'eliminazione di un ruolo o di un profilo di istanza associato a un'istanza in esecuzione interrompe tutte le applicazioni in esecuzione sull'istanza.

Se si preferisce non eliminare definitivamente un ruolo, è possibile disabilitarlo. A tale scopo, modifica le policy del ruolo e quindi revoca tutte le sessioni correnti. Ad esempio, potresti aggiungere una policy al ruolo che nega l'accesso a tutti. AWSÈ inoltre possibile modificare i criteri di attendibilità per negare l'accesso a tutti coloro che tentano di assumere il ruolo. Per ulteriori informazioni sull'avvio delle sessioni, consulta Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

Argomenti

- Visualizzazione dell'accesso per il ruolo
- Eliminazione del ruolo collegato ai servizi
- Eliminazione di un ruolo IAM (console)

- Eliminazione di un ruolo IAM (AWS CLI)
- Eliminazione di un ruolo IAM (API AWS)
- Informazioni correlate

Visualizzazione dell'accesso per il ruolo

Prima di eliminare un ruolo, è opportuno esaminare quando è stato utilizzato per l'ultima volta. È possibile eseguire questa operazione utilizzando l' AWS Management Console AWS CLI, l'o l' AWS API. È consigliabile visualizzare queste informazioni per non privare dell'accesso qualcuno che utilizza il ruolo.

La data dell'ultima attività del ruolo potrebbe non corrispondere all'ultima data segnalata nella scheda Access Advisor. La scheda Access Advisor riporta l'attività solo per i servizi consentiti dalle policy di autorizzazione del ruolo. La data dell'ultima attività del ruolo include l'ultimo tentativo di accesso a qualsiasi servizio in AWS.

> **Note**
>
> Il periodo di monitoraggio dei dati sull'ultima attività di un ruolo e di Access Advisor include gli ultimi 400 giorni. Questo periodo può essere abbreviato se la regione ha iniziato a supportare queste funzionalità nell'ultimo anno. Il ruolo potrebbe essere stato utilizzato più di 400 giorni fa. Per ulteriori informazioni sul periodo di monitoraggio, consulta Dove AWS tiene traccia delle ultime informazioni a cui si accede.

Per visualizzare la data di ultimo utilizzo di un ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Individua la riga del ruolo con l'attività che si desidera visualizzare. È possibile utilizzare il campo di ricerca per restringere i risultati. Visualizzare la colonna Last activity (Ultima attività) per visualizzare il numero di giorni trascorsi dalla data di ultimo utilizzo del ruolo. Se il ruolo non è stato utilizzato entro il periodo di monitoraggio, nella tabella viene visualizzato None (Nessuno).
4. Scegliere il nome del ruolo per visualizzare ulteriori informazioni. La pagina Riepilogo del ruolo include anche Ultima attività, che visualizza la data dell'ultimo utilizzo del ruolo. Se il ruolo non è

stato utilizzato negli ultimi 400 giorni, Last activity (Ultima attività) visualizza Not accessed in the tracking period (Nessun accesso nel periodo di monitoraggio).

Per visualizzare la data di ultimo utilizzo di un ruolo (AWS CLI)

`aws iam get-role` - Eseguire questo comando per restituire le informazioni su un ruolo, incluso l'oggetto `RoleLastUsed`. Questo oggetto contiene `LastUsedDate` e la `Region` in cui il ruolo è stato utilizzato per l'ultima volta. Se `RoleLastUsed` è presente ma non contiene un valore, il ruolo non è stato utilizzato entro il periodo di monitoraggio.

Per visualizzare quando un ruolo è stato utilizzato l'ultima volta (AWS API)

`GetRole` - Chiamare questa operazione per restituire le informazioni su un ruolo, incluso l'oggetto `RoleLastUsed`. Questo oggetto contiene `LastUsedDate` e la `Region` in cui il ruolo è stato utilizzato per l'ultima volta. Se `RoleLastUsed` è presente ma non contiene un valore, il ruolo non è stato utilizzato entro il periodo di monitoraggio.

Eliminazione del ruolo collegato ai servizi

Se il ruolo è un ruolo collegato al servizio, consulta la documentazione del servizio collegato per ulteriori informazioni su come eliminare il ruolo. Puoi visualizzare i ruoli collegati ai servizi nell'account visitando la pagina Ruoli IAM nella console. Per i ruoli collegati al servizio viene visualizzata l'indicazione (Service-linked role) (Ruolo collegato al servizio) nella colonna Trusted entities (Entità attendibili) della tabella. Un banner nella pagina Riepilogo del ruolo indica anche che il ruolo è un ruolo collegato ai servizi.

Se il servizio non include la documentazione per l'eliminazione del ruolo collegato al servizio, puoi utilizzare la console IAM o l'API per eliminare il ruolo. AWS CLI Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi.

Eliminazione di un ruolo IAM (console)

Quando utilizzi il AWS Management Console per eliminare un ruolo, IAM scollega automaticamente le policy gestite associate al ruolo. Inoltre elimina automaticamente anche le policy in linea associate al ruolo e qualsiasi profilo dell'istanza Amazon EC2 che contiene il ruolo.

> **Important**
>
> In alcuni casi, un ruolo potrebbe essere associato a un profilo dell'istanza Amazon EC2 e il ruolo e il profilo dell'istanza potrebbero avere lo stesso nome. In tal caso puoi utilizzare il

AWS Management Console per eliminare il ruolo e il profilo dell'istanza. Questo collegamento avviene automaticamente per i ruoli e i profili delle istanze creati nella console. Se hai creato il ruolo da AWS CLI, Tools for Windows PowerShell o dall' AWS API, il ruolo e il profilo dell'istanza potrebbero avere nomi diversi. In questo caso non è possibile utilizzare la console per eliminarli. È invece necessario utilizzare Tools for Windows PowerShell o AWS API per rimuovere prima il ruolo dal profilo dell'istanza. AWS CLIÈ quindi necessario eseguire un passaggio distinto per eliminare il ruolo.

Per eliminare un ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegliere Roles (Ruoli), quindi selezionare la casella di controllo accanto al nome del ruolo che si desidera eliminare.
3. Nella parte superiore della pagina, scegli Elimina.
4. Nella finestra di dialogo di conferma, esamina le informazioni relative all'ultimo accesso, che mostrano l'ultima volta che ciascuno dei ruoli selezionati ha effettuato l'ultimo accesso a un AWS servizio. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, inserisci il nome del ruolo nel campo di immissione testo e seleziona Elimina. Se sei sicuro, puoi procedere con l'eliminazione anche se l'ultimo accesso ai dati del servizio è ancora in fase di caricamento.

> **Note**
>
> Non è possibile utilizzare la console per eliminare un profilo dell'istanza, a meno che non abbia lo stesso nome del ruolo. Il profilo dell'istanza viene eliminato come parte del processo di eliminazione di un ruolo descritto nella procedura precedente. Per eliminare un profilo di istanza senza eliminare anche il ruolo, è necessario utilizzare l' AWS API AWS CLI o. Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Eliminazione di un ruolo IAM (AWS CLI)

Quando si utilizza il AWS CLI per eliminare un ruolo, è necessario innanzitutto eliminare le politiche in linea associate al ruolo. È inoltre necessario scollegare le policy gestite associate al ruolo. Se desideri eliminare il profilo dell'istanza associato che contiene il ruolo, devi eliminarlo separatamente.

Per eliminare un ruolo (AWS CLI)

1. Se non conosci il nome del ruolo da eliminare, immetti il comando seguente per elencare i ruoli nell'account:

   ```
   aws iam list-roles
   ```

   L'elenco include l'Amazon Resource Name (ARN) di ogni ruolo. Per fare riferimento ai ruoli con i comandi della CLI utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.

2. Rimuovi il ruolo da tutti i profili delle istanze a cui è associato.

   a. Per elencare tutti i profili delle istanze cui è associato il ruolo, immetti il seguente comando:

      ```
      aws iam list-instance-profiles-for-role --role-name role-name
      ```

   b. Per rimuovere il ruolo da un profilo dell'istanza, immetti il seguente comando per ogni profilo dell'istanza:

      ```
      aws iam remove-role-from-instance-profile --instance-profile-name instance-profile-name --role-name role-name
      ```

3. Elimina tutte le policy associate al ruolo.

   a. Per elencare tutte le policy inline presenti nel ruolo, immetti il seguente comando:

      ```
      aws iam list-role-policies --role-name role-name
      ```

   b. Per eliminare ogni policy inline dal ruolo, immetti il seguente comando per ogni policy:

      ```
      aws iam delete-role-policy --role-name role-name --policy-name policy-name
      ```

   c. Per elencare tutte le policy gestite collegate al ruolo, immetti il seguente comando:

      ```
      aws iam list-attached-role-policies --role-name role-name
      ```

   d. Per scollegare ogni policy gestita dal ruolo, immetti il seguente comando per ogni policy:

      ```
      aws iam detach-role-policy --role-name role-name --policy-arn policy-arn
      ```

4. Immetti il seguente comando per eliminare il ruolo:

```
aws iam delete-role --role-name role-name
```

5. Se non prevedi di riutilizzare i profili delle istanze associati al ruolo, puoi immettere il seguente comando per eliminarli:

```
aws iam delete-instance-profile --instance-profile-name instance-profile-name
```

Eliminazione di un ruolo IAM (API AWS)

Se utilizzi l'API IAM per eliminare un ruolo, devi prima eliminare le policy inline associate al ruolo. È inoltre necessario scollegare le policy gestite associate al ruolo. Se desideri eliminare il profilo dell'istanza associato che contiene il ruolo, devi eliminarlo separatamente.

Per eliminare un ruolo (AWS API)

1. Per elencare tutti i profili di istanza a cui è associato un ruolo, chiama ListInstanceProfilesForRole.

   Per rimuovere il ruolo da un profilo di istanza, chiama RemoveRoleFromInstanceProfile. È necessario passare il nome del ruolo e il nome del profilo di istanza.

   Se non intendi riutilizzare un profilo di istanza associato al ruolo, chiama DeleteInstanceProfileper eliminarlo.
2. Per elencare tutte le politiche in linea per un ruolo, chiama. ListRolePolicies

   Per eliminare le politiche in linea associate al ruolo, chiama. DeleteRolePolicy Devi passare il nome del ruolo e il nome della policy inline.
3. Per elencare tutte le politiche gestite associate a un ruolo, chiama ListAttachedRolePolicies.

   Per scollegare le politiche gestite allegate al ruolo, chiama DetachRolePolicy. Devi passare il nome del ruolo e l'ARN della policy gestita.
4. Chiama DeleteRoleper eliminare il ruolo.

Informazioni correlate

Per informazioni generali sui profili delle istanze, consulta Utilizzo dei profili delle istanze.

Per informazioni generali sui ruoli collegati al servizio, consultare Uso di ruoli collegati ai servizi.

Provider di identità e federazione

Se gestisci già le identità degli utenti all'esterno di AWS, puoi utilizzare i provider di identità invece di creare utenti IAM nel tuo. Account AWS Con un provider di identità (IdP), puoi gestire le tue identità utente all'esterno AWS e concedere a queste identità utente esterne le autorizzazioni per utilizzare AWS le risorse del tuo account. Questa funzione è utile se la tua organizzazione dispone già di un proprio sistema di gestione delle identità, come ad esempio una directory aziendale degli utenti. Risulta utile anche quando devi creare un'app per dispositivi mobili o un'applicazione Web che richieda l'accesso alle risorse AWS .

Un IdP esterno fornisce informazioni sull'identità AWS utilizzando OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). OIDC collega applicazioni, come GitHub Actions, che non funzionano su risorse. AWS AWS Esempi di noti provider di identità SAML sono Shibboleth e Active Directory Federation Services.

> **Note**
>
> Come best practice di sicurezza, consigliamo di gestire gli utenti umani in Centro identità IAM con un gestore dell'identità digitale SAML esterno anziché utilizzare la federazione SAML in IAM. Per informazioni su situazioni specifiche in cui è richiesto un utente IAM, consulta la sezione Quando creare un utente IAM invece di un ruolo.

Quando utilizzi un provider di identità, non devi creare un codice di accesso personalizzato né gestire le tue identità utente. Tale operazione viene eseguita dall'IdP. I tuoi utenti esterni accedono tramite un IdP e puoi concedere a tali identità esterne le autorizzazioni per utilizzare le AWS risorse del tuo account. I provider di identità aiutano a mantenere la tua Account AWS sicurezza perché non devi distribuire o incorporare credenziali di sicurezza a lungo termine, come le chiavi di accesso, nell'applicazione.

Questa guida illustra la federazione IAM. Il tuo caso d'uso potrebbe essere più adatto al Centro identità IAM o ad Amazon Cognito. I riepiloghi e la tabella seguenti forniscono una panoramica dei metodi che gli utenti possono utilizzare per ottenere l'accesso federato alle risorse. AWS

	Tipo di account	Gestione degli accessi di...	Origine di identità supportata
Federazione con il Centro identità IAM	Account multipli gestiti da AWS Organizations	Utenti umani della forza lavoro	• SAML 2.0 • Active Directory gestita • Directory del Centro identità
Federazione con IAM	Singolo account autonomo	• Utenti umani impegnati in implementazioni a breve termine su piccola scala • Utenti di macchine	• SAML 2.0 • OIDC
Federazione con pool di identità Amazon Cognito	Qualsiasi	Utenti di app che richiedono l'autorizzazione IAM per accedere alle risorse	• SAML 2.0 • OIDC • Selezionare un gestore di identità digitali social OAuth 2.0

Federazione con il Centro identità IAM

Per una gestione centralizzata degli accessi degli utenti umani, consigliamo di utilizzare Centro identità IAM per gestire l'accesso ai tuoi account e le autorizzazioni all'interno di questi account. Agli utenti di IAM Identity Center vengono concesse credenziali a breve termine per AWS le tue risorse. Puoi utilizzare Active Directory, un provider di identità (IdP) esterno o una directory IAM Identity Center come origine di identità per utenti e gruppi per assegnare l'accesso alle tue risorse. AWS

IAM Identity Center supporta la federazione delle identità con SAML (Security Assertion Markup Language) 2.0 per fornire un accesso single sign-on federato agli utenti autorizzati a utilizzare le applicazioni all'interno del portale di accesso. AWS Gli utenti possono quindi accedere ai servizi che supportano SAML, incluse le applicazioni AWS Management Console e quelle di terze parti, come Microsoft 365, SAP Concur e Salesforce.

Federazione con IAM

Anche se consigliamo vivamente di gestire gli utenti umani nel Centro identità IAM, è possibile abilitare l'accesso agli utenti federati con IAM per gli utenti umani impegnati in implementazioni a breve termine su piccola scala. IAM consente di utilizzare SAML 2.0 e Open ID Connect (OIDC) separati IdPs e di utilizzare attributi utente federati per il controllo degli accessi. Con IAM, puoi trasferire gli attributi utente, come centro di costo, titolo o locale, dal tuo IdPs a e implementare autorizzazioni di AWS accesso granulari basate su questi attributi.

Un carico di lavoro è una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione o un processo back-end. Il tuo carico di lavoro può richiedere un'identità IAM per effettuare richieste a AWS servizi, applicazioni, strumenti operativi e componenti. Queste identità includono macchine in esecuzione nei tuoi AWS ambienti, come AWS Lambda istanze o funzioni di Amazon EC2.

Puoi anche gestire identità computer per soggetti esterni che necessitano di accesso. Per concedere l'accesso alle identità computer, puoi utilizzare i ruoli IAM. I ruoli IAM dispongono di autorizzazioni specifiche e forniscono un modo per accedere AWS affidandosi a credenziali di sicurezza temporanee con una sessione di ruolo. Inoltre, potresti avere macchine esterne AWS che richiedono l'accesso ai tuoi ambienti. AWS Per le macchine che funzionano all'esterno dell' AWS utente, puoi utilizzare IAM Roles Anywhere. Per ulteriori informazioni sui ruoli, consulta Ruoli IAM. Per i dettagli su come utilizzare i ruoli per delegare l'accesso da una parte all'altra Account AWS, consultaTutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.

Per collegare un IdP direttamente a IAM, crei un'entità provider di identità per stabilire una relazione di fiducia tra il tuo Account AWS e l'IdP. Supporti IdPs IAM compatibili con OpenID Connect (OIDC) o SAML 2.0 (Security Assertion Markup Language 2.0). Per ulteriori informazioni sull'utilizzo di uno di questi IdPs con AWS, consulta le seguenti sezioni:

- Federazione OIDC
- Federazione SAML 2.0

Federazione con pool di identità Amazon Cognito

Amazon Cognito è progettato per gli sviluppatori che desiderano autenticare e autorizzare gli utenti nelle proprie app mobili e Web. I pool di utenti di Amazon Cognito aggiungono funzionalità di accesso e registrazione all'app e i pool di identità forniscono credenziali IAM che consentono agli utenti di

accedere alle risorse protette gestite in AWS. I pool di identità acquisiscono le credenziali per le sessioni temporanee tramite il funzionamento dell'operazione API `AssumeRoleWithWebIdentity`.

Amazon Cognito funziona con gestori dell'identità digitale esterni che supportano SAML e OpenID Connect e con gestori di identità social come Facebook, Google e Amazon. L'app può far effettuare l'accesso a un utente con un pool di utenti o un gestore dell'identità digitale esterno e recuperare in seguito le risorse per suo conto con sessioni temporanee personalizzate con un ruolo IAM.

Scenari comuni

> **Note**
>
> Ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso AWS. Hai preso in considerazione l'utilizzo AWS IAM Identity Center? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli utenti un accesso Single Sign-On protetto da MFA a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Puoi utilizzare un provider di identità esterno (IdP) per gestire le identità degli utenti al di fuori di AWS. e l'IdP esterno. Un IdP esterno può fornire informazioni sull'identità AWS utilizzando OpenID Connect (OIDC) o Security Assertion Markup Language (SAML). L'OIDC viene comunemente utilizzato quando un'applicazione che non funziona richiede l'accesso alle risorse. AWS AWS

Quando desideri configurare la federazione con un IdP esterno, crei un provider di identità IAM per fornire AWS informazioni sull'IdP esterno e sulla sua configurazione. In questo modo si instaura un rapporto di fiducia tra il tuo Account AWS e l'IdP esterno. I seguenti argomenti forniscono scenari comuni per l'utilizzo dei provider di identità IAM.

Argomenti

- Utilizzo di Amazon Cognito per applicazioni per dispositivi mobili
- Utilizzo delle operazioni dell'API di federazione OIDC per app mobili

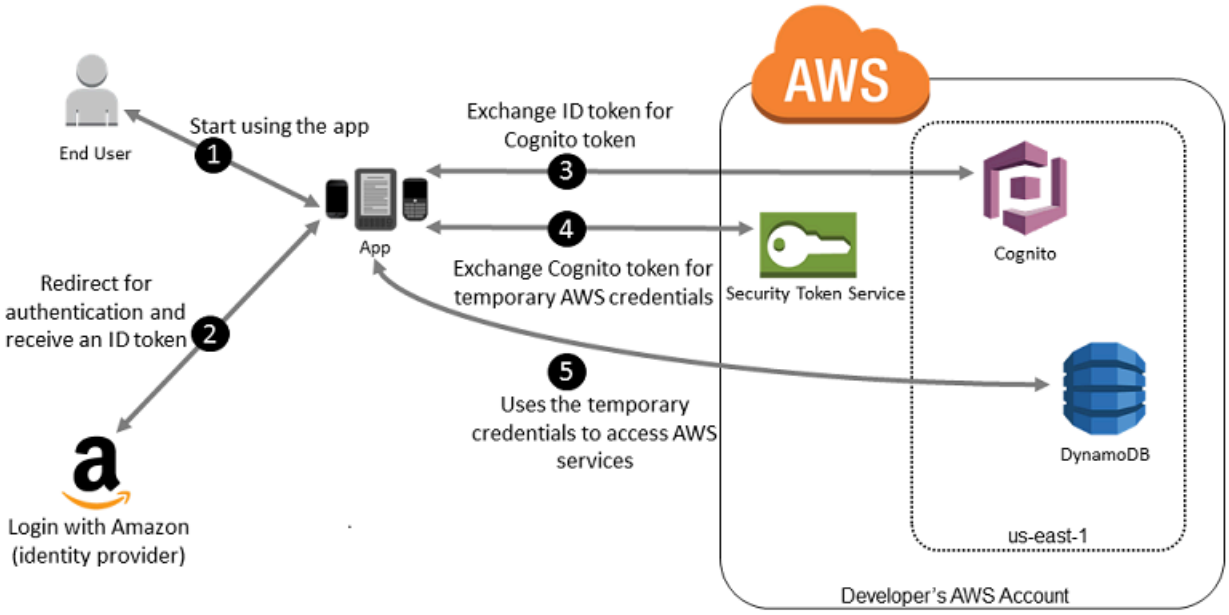
Utilizzo di Amazon Cognito per applicazioni per dispositivi mobili

Il modo preferito per utilizzare la federazione OIDC è usare Amazon Cognito. Ad esempio, Adele sta sviluppando un gioco per un dispositivo mobile in cui i dati dell'utente, quali punteggi e profili, vengono memorizzati in Amazon S3 e Amazon DynamoDB. Adele potrebbe archiviare questi dati anche in locale sul dispositivo e utilizzare Amazon Cognito per mantenerli sincronizzati su tutti i dispositivi. Tuttavia, Adele è consapevole che, per motivi di sicurezza e manutenzione, le credenziali di sicurezza AWS a lungo termine non dovrebbero essere distribuite con il gioco. Adele sa anche che il gioco potrebbe avere un gran numero di utenti. Per tutti questi motivi, non desidera creare nuove identità utente in IAM per ciascun giocatore. In alternativa, decide di sviluppare il gioco in modo che gli utenti possano effettuare l'accesso utilizzando un'identità già stabilita con un noto provider di identità (IdP) esterno, ad esempio Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Il suo gioco può sfruttare il meccanismo di autenticazione di uno di questi provider per convalidare l'identità dell'utente.

Per consentire all'app mobile di accedere alle sue AWS risorse, Adele si registra innanzitutto per ottenere un ID sviluppatore con il nome scelto. IdPs Configura anche l'applicazione con ciascuno di questi provider. Nella cartella Account AWS che contiene il bucket Amazon S3 e la tabella DynamoDB per il gioco, Adele utilizza Amazon Cognito per creare ruoli IAM che definiscono con precisione le autorizzazioni di cui il gioco ha bisogno. Se utilizza un IdP OIDC, crea anche un'entità provider di identità IAM OIDC per stabilire un rapporto di fiducia tra il pool di identità di Amazon Cognito che possiede e l'IdP. Account AWS

Nel codice dell'app, Adele chiama l'interfaccia di accesso per il provider di identità che ha configurato in precedenza. Il provider di identità gestisce tutti i dettagli relativi all'accesso dell'utente e l'app riceve un token di accesso OAuth o un token ID OIDC dal provider. L'app di Adele può scambiare queste informazioni di autenticazione con una serie di credenziali di sicurezza temporanee costituite da un ID della chiave di accesso, una chiave di AWS accesso segreta e un token di sessione. L'app può quindi utilizzare queste credenziali per accedere ai servizi web offerti da. AWS L'app è limitata alle autorizzazioni definite nel ruolo che assume.

La figura riportata di seguito mostra un flusso semplificato su come questo processo potrebbe funzionare, usando Login with Amazon come provider di identità. Per la fase 2, l'app può anche utilizzare Facebook, Google o qualsiasi altro IdP compatibile con OIDC, ma ciò non è mostrato qui.



1. Un cliente avvia l'app su un dispositivo mobile. L'app richiede all'utente di effettuare l'accesso.
2. L'app utilizza il login con le risorse di Login with Amazon per accettare le credenziali dell'utente.
3. L'app utilizza le operazioni dell'API Amazon Cognito `GetId` e `GetCredentialsForIdentity` per scambiare il token ID di Login with Amazon con un token di Amazon Cognito. Amazon Cognito, che è stato configurato per rendere attendibile il tuo progetto Login with Amazon, genera un token che scambia con credenziali di sessione temporanee per AWS STS.
4. L'app riceve le credenziali di sicurezza temporanee da Amazon Cognito. La tua app può anche utilizzare il flusso di lavoro Basic (Classic) in Amazon Cognito per recuperare i token da utilizzare. AWS STS `AssumeRoleWithWebIdentity` Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità (identità federate) nella Guida per gli sviluppatori di Amazon Cognito.
5. Le credenziali di sicurezza temporanee possono essere utilizzate dall'app per accedere alle risorse AWS richieste dall'applicazione per funzionare. Il ruolo associato alle credenziali di sicurezza temporanee e alle relative policy assegnate determina a quali elementi è possibile accedere.

Utilizza la seguente procedura per configurare la tua app in modo che utilizzi Amazon Cognito per autenticare gli utenti e consentire all'app di accedere alle risorse. AWS Per le operazioni specifiche per realizzare questo scenario, consulta la documentazione di Amazon Cognito.

1. (Facoltativo) Registrati come sviluppatore con Login with Amazon, Facebook, Google o qualsiasi altro provider di identità compatibile con OpenID Connect (OIDC) e configura una o più app

con il provider. Questa fase è facoltativa poiché Amazon Cognito supporta anche l'accesso non autenticato (guest) per gli utenti.

2. Vai ad Amazon Cognito in. AWS Management Console Utilizza la procedura guidata di Amazon Cognito per creare un pool di identità, ovvero un container che Amazon Cognito utilizza per mantenere le identità degli utenti finali organizzate per le app. Puoi condividere i pool di identità tra le app. Quando configuru un pool di identità, Amazon Cognito crea uno o due ruoli IAM (uno per le identità autenticate e uno per le identità "guest" non autenticate) che definiscono le autorizzazioni per gli utenti di Amazon Cognito.
3. Integra AWSAmplify con l'app e importa i file necessari per utilizzare Amazon Cognito.
4. Crea un'istanza del provider di credenziali di Amazon Cognito, passando l'ID del pool di identità, il numero del tuo Account AWS e l'Amazon Resource Name (ARN) dei ruoli associati al pool di identità. La procedura guidata di Amazon Cognito AWS Management Console fornisce un codice di esempio per aiutarti a iniziare.
5. Quando l'app accede a una AWS risorsa, passa l'istanza del provider di credenziali all'oggetto client, che passa le credenziali di sicurezza temporanee al client. Le autorizzazioni per le credenziali si basano sul ruolo o sui ruoli definiti in precedenza.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Accedi (Android) nella documentazione del AWS Amplify Framework.
- Accedi (iOS) nella documentazione del AWS Amplify Framework.

Utilizzo delle operazioni dell'API di federazione OIDC per app mobili

Per ottenere i migliori risultati, usa Amazon Cognito come broker di identità per quasi tutti gli scenari di federazione OIDC. Amazon Cognito è semplice da utilizzare e fornisce funzionalità aggiuntive quali l'accesso anonimo (non autenticato) e la sincronizzazione dei dati degli utenti tra più dispositivi e provider. Tuttavia, se hai già creato un'app che utilizza la federazione OIDC chiamando manualmente l'`AssumeRoleWithWebIdentity`API, puoi continuare a usarla e le tue app continueranno a funzionare correttamente.

Il processo per utilizzare la federazione OIDC senza Amazon Cognito segue questo schema generale:

1. Effettuare la registrazione come sviluppatore al provider di identità (IdP) esterno e configurare l'app con tale provider, che fornisce un ID univoco per l'app. (Diversi IdPs utilizzano una terminologia

diversa per questo processo. Questo schema utilizza il termine configura per il processo di identificazione dell'app con l'IdP.) Ogni IdP ti fornisce un ID app univoco per quell'IdP, quindi se configuri la stessa app con più ID IdPs, la tua app avrà più ID app. È possibile configurare più app con ciascun provider.

I seguenti link esterni forniscono informazioni sull'utilizzo di alcuni dei provider di identità più utilizzati ()IdPs:

- Centro Sviluppatori di Login with Amazon
- Aggiunta dell'accesso a Facebook a un'app o a un sito Web sul sito degli sviluppatori di Facebook.
- Utilizzo di OAuth 2.0 per l'accesso (OpenID Connect) sul sito degli sviluppatori di Google.

> **Important**
>
> Se utilizzi un provider di identità OIDC di Google, Facebook o Amazon Cognito, non creare un provider di identità IAM separato in. AWS Management Console AWS dispone di questi provider di identità OIDC integrati e disponibili per l'uso da parte tua. Ignora la fase seguente e passa direttamente alla creazione di nuovi ruoli utilizzando il provider di identità.

2. Se utilizzi un IdP di identità diverso da Google, Facebook o Amazon Cognito compatibile con OIDC, crea un'entità provider di identità IAM.
3. In IAM, crea uno o più ruoli. Per ogni ruolo, definisci chi può assumere il ruolo (policy di attendibilità) e quali autorizzazioni concedere agli utenti dell'app (policy di autorizzazione). Di solito, è necessario creare un ruolo per ogni provider di identità supportato da un'app. Ad esempio, puoi creare un ruolo che può essere assunto da un'applicazione quando l'utente effettua l'accesso tramite Login with Amazon, un secondo ruolo per la stessa applicazione in cui l'utente effettua l'accesso tramite Facebook e un terzo ruolo per l'applicazione in cui l'utente effettua l'accesso tramite Google. Per la relazione di trust, specificare il provider di identità (ad esempio Amazon.com) come `Principal` (l'entità attendibile) e includere un elemento `Condition` corrispondente all'ID app assegnato dal provider di identità. Esempi di ruoli per diversi provider sono descritti più in Creazione di un ruolo per un provider di identità di terza parte (federazione).
4. Nell'applicazione, autenticare gli utenti con il provider di identità. Le specifiche della procedura variano sia in base al provider di identità in uso (Login with Amazon, Facebook o Google) sia in base alla piattaforma su cui viene eseguita l'app. Ad esempio, il metodo di autenticazione di un'app Android può differire da quello di un'app iOS o di un'app Web JavaScript basata.

In genere, se l'utente non ha già effettuato l'accesso, il provider di identità si occupa di visualizzare una pagina di accesso. Dopo aver autenticato l'utente, il provider di identità restituisce all'app un token di autenticazione con le informazioni sull'utente. Le informazioni incluse dipendono dagli elementi esposti dal provider di identità e dalle informazioni che l'utente è disposto a condividere. Queste informazioni possono essere utilizzate nell'app.

5. Nell'app, effettuare una chiamata non firmata all'operazione `AssumeRoleWithWebIdentity` per richiedere le credenziali di sicurezza provvisorie. Nella richiesta, passi il token di autenticazione dell'IdP e specifichi l'Amazon Resource Name (ARN) per il ruolo IAM che hai creato per quell'IdP. AWS verifica che il token sia affidabile e valido e, in tal caso, restituisce all'app credenziali di sicurezza temporanee che dispongono delle autorizzazioni per il ruolo indicato nella richiesta. La risposta include anche i metadati relativi all'utente forniti dal provider di identità, ad esempio l'ID utente univoco che il provider associa all'utente.
6. Utilizzando le credenziali di sicurezza temporanee della `AssumeRoleWithWebIdentity` risposta, l'app invia richieste firmate alle operazioni API. AWS Le informazioni sull'ID utente fornite dall'IdP possono distinguere gli utenti nella tua app. Ad esempio, puoi inserire oggetti nelle cartelle Amazon S3 che includono l'ID utente come prefissi o suffissi. Ciò consente di creare policy di controllo degli accessi che bloccano la cartella in modo che solo l'utente con l'ID specificato possa accedervi. Per ulteriori informazioni, consulta AWS STS principi di sessione utente federati.
7. L'app dovrebbe memorizzare nella cache le credenziali di sicurezza temporanee in modo da non doverne ottenere di nuove ogni volta che ha bisogno di effettuare una richiesta ad AWS. Come impostazione predefinita, le credenziali sono valide per un'ora. Quando scadono (o prima), devi effettuare un'altra chiamata ad `AssumeRoleWithWebIdentity` per ottenere un nuovo set di credenziali di sicurezza temporanee. A seconda del provider di identità e di come gestisce i token, potrebbe essere necessario aggiornare il token del provider prima di effettuare una nuova chiamata ad `AssumeRoleWithWebIdentity`, dato che anche i token di solito scadono dopo un determinato periodo di tempo. Se utilizzi l' AWS SDK per iOS o l'SDK per Android, puoi utilizzare AWS l'azione CredentialsProvider AmazonSTS, che gestisce le credenziali temporanee IAM, incluso l'aggiornamento delle stesse come richiesto.

Federazione OIDC

Immagina di creare un'applicazione che accede a AWS risorse, come GitHub Actions che utilizza flussi di lavoro per accedere ad Amazon S3 e DynamoDB.

Quando utilizzi questi flussi di lavoro, effettui richieste ai AWS servizi che devono essere firmate con una chiave di accesso. AWS Tuttavia, consigliamo vivamente di non archiviare AWS le credenziali a lungo termine in applicazioni esterne. AWSConfigura invece le tue applicazioni per richiedere le credenziali di AWS sicurezza temporanee in modo dinamico quando necessario utilizzando la federazione OIDC. Le credenziali temporanee fornite sono mappate a un AWS ruolo che dispone solo delle autorizzazioni necessarie per eseguire le attività richieste dall'applicazione.

Con la federazione OIDC, non è necessario creare un codice di accesso personalizzato o gestire le proprie identità utente. Puoi invece utilizzare OIDC in applicazioni, come GitHub Actions o qualsiasi altro IdP compatibile con OpenID Connect (OIDC), con cui effettuare l'autenticazione. AWS Ricevono un token di autenticazione, noto come JSON Web Token (JWT), e quindi lo scambiano con credenziali di sicurezza temporanee in AWS quella mappa con un ruolo IAM con le autorizzazioni per utilizzare risorse specifiche dell'utente. Account AWS L'utilizzo di un IdP ti aiuta a mantenere la tua Account AWS sicurezza perché non devi incorporare e distribuire credenziali di sicurezza a lungo termine con l'applicazione.

Per la maggior parte degli scenari, consigliamo di utilizzare Amazon Cognito in quanto agisce come gestore identità e svolge la maggior parte delle attività di federazione per tuo conto. Per informazioni dettagliate, consultare la sezione seguente, Utilizzo di Amazon Cognito per applicazioni per dispositivi mobili.

> **Note**
>
> I JSON Web Tokens (JWT) emessi dai provider di identità OpenID Connect (OIDC) contengono una data di scadenza nell'attestazione che specifica quando scade il `exp` token. IAM offre una finestra di cinque minuti oltre la data di scadenza specificata nel JWT per tenere conto dell'inclinazione dell'orologio, come consentito dallo standard OpenID Connect (OIDC) Core 1.0. Ciò significa che i JWT OIDC ricevuti da IAM dopo la scadenza ma entro questo intervallo di cinque minuti vengono accettati per un'ulteriore valutazione ed elaborazione.

Argomenti

- Creare un provider di identità OpenID Connect (OIDC) in IAM
- Ottieni l'impronta personale di un provider di identità OpenID Connect
- Risorse aggiuntive per la federazione OIDC

Creare un provider di identità OpenID Connect (OIDC) in IAM

I provider di identità OIDC IAM sono entità in IAM che descrivono un servizio del provider di identità (IdP) esterno in grado di supportare lo standard OpenID Connect (OIDC), come Google o Salesforce. È possibile utilizzare un provider di identità OIDC IAM per stabilire la fiducia tra un provider di identità compatibile con OIDC e il tuo Account AWS. È utile quando si crea un'app mobile o un'applicazione web che richiede l'accesso alle AWS risorse, ma non si desidera creare un codice di accesso personalizzato o gestire le proprie identità utente. Per ulteriori informazioni su questo scenario, consulta the section called “federazione OIDC”.

Puoi creare e gestire un provider di identità IAM OIDC utilizzando l' AWS Management Console AWS Command Line Interface API Tools for Windows PowerShell o IAM.

Una volta creato un provider di identità OIDC IAM, dovrai creare uno o più ruoli IAM. Un ruolo è un'identità AWS che non ha le proprie credenziali (come invece fa un utente). Tuttavia, in questo contesto, un ruolo viene assegnato in modo dinamico a un utente federato autenticato dall'IdP dell'organizzazione. Il ruolo consente al provider di identità dell'organizzazione di richiedere credenziali di sicurezza provvisorie per l'accesso a AWS. Le politiche assegnate al ruolo determinano le operazioni consentite agli utenti federati. AWS Per creare un ruolo per un provider di identità di terze parti, consulta Creazione di un ruolo per un provider di identità di terza parte (federazione).

> **Important**
>
> Quando si configurano politicy basate sull'identità per operazioni che supportano risorse `oidc-provider`, IAM valuta l'URL completo del provider di identità OIDC, inclusi i percorsi specificati. Se l'URL del tuo provider di identità OIDC ha un percorso, devi includere quel percorso nell’ARN `oidc-provider` come valore dell'elemento `Resource`. È inoltre possibile aggiungere una barra in avanti e un carattere jolly (`/*`) al dominio URL o usare caratteri jolly (`*` e `?`) in qualsiasi punto del percorso dell'URL. Se l'URL del provider di identità OIDC nella richiesta non corrisponde al valore impostato nell’elemento `Resource` della policy, la richiesta ha esito negativo.

Per risolvere i problemi più comuni relativi alla federazione IAM OIDC, consulta Risolvere gli errori relativi a OIDC su re:POST. AWS

Argomenti

- Prerequisiti: convalida la configurazione del tuo provider di identità

- Creazione e gestione di un provider OIDC (Console)
- Creazione e gestione di un provider di identità OIDC IAM (AWS CLI)
- Creazione e gestione di un provider di identità (API) OIDC AWS

Prerequisiti: convalida la configurazione del tuo provider di identità

Prima di poter creare un provider di identità IAM OIDC, è necessario disporre delle seguenti informazioni dal proprio IdP. Per ulteriori informazioni su come ottenere informazioni sulla configurazione del provider OIDC, consulta la documentazione del tuo IdP.

1. Determina l'URL disponibile pubblicamente del tuo provider di identità OIDC. L'URL deve iniziare con https://. Secondo lo standard OIDC, i componenti del percorso sono consentiti ma i parametri di interrogazione no. In genere, l'URL è composto solo da un nome host, ad esempio https://server.example.org o https://example.com. L'URL non deve contenere un numero di porta.
2. Aggiungi /.well-known/openid-configuration alla fine dell'URL del tuo provider di identità OIDC per visualizzare il documento di configurazione e i metadati disponibili pubblicamente del provider. È necessario disporre di un documento di rilevamento in formato JSON con il documento di configurazione e i metadati del provider che possono essere recuperati dall'URL dell'endpoint di scoperta del provider OpenID Connect.
3. Verifica che i seguenti valori siano inclusi nelle informazioni di configurazione del tuo provider. Se nella configurazione openid manca uno di questi campi, è necessario aggiornare il documento di scoperta. Questo processo può variare in base al provider di identità, quindi segui la documentazione del tuo IdP per completare questa attività.

    - issuer: l'URL del tuo dominio.
    - jwks_uri: l'endpoint JSON Web Key Set (JWKS) da cui IAM ottiene le tue chiavi pubbliche. Il tuo provider di identità deve includere un endpoint JSON Web Key Set (JWKS) nella configurazione openid. Questo URI definisce dove ottenere le chiavi pubbliche utilizzate per verificare i token firmati dal provider di identità.
    - claims_supported: informazioni sull'utente che ti aiutano a garantire che le risposte di autenticazione OIDC del tuo IdP contengano gli attributi richiesti AWS utilizzati nelle policy IAM per verificare le autorizzazioni per gli utenti federati. Per un elenco delle chiavi di condizione IAM che possono essere utilizzate per le attestazioni, consulta. Chiavi disponibili per la federazione AWS OIDC
        - aud: devi determinare il valore dichiarato del pubblico dal tuo IdP in JSON Web Tokens (JWT). L'attestazione audience (aud) è specifica dell'applicazione e identifica i destinatari

previsti del token. Quando registri un'app mobile o web con un provider OpenID Connect, viene stabilito un ID client che identifica l'applicazione. L'ID client è un identificatore univoco per l'app che viene trasmesso nel reclamo di autenticazione aud. Il claim aud deve corrispondere al valore Audience quando crei il tuo provider di identità IAM OIDC.

  - iat: i reclami devono includere un valore `iat` che rappresenti l'ora in cui viene emesso il token ID.
  - iss: l'URL del provider di identità. L'URL deve iniziare con https://e deve corrispondere all'URL del provider fornito a IAM. Secondo lo standard OIDC, i componenti del percorso sono consentiti ma i parametri di query no. In genere, l'URL è composto solo da un nome host, ad esempio https://server.example.org o https://example.com. L'URL non deve contenere un numero di porta.
- response_types_supported: id_token
- subject_types_supported: pubblico
- id_token_signing_alg_values_supported: RS256

> **Note**
>
> Puoi includere rivendicazioni aggiuntive come quelle personalizzate nell'esempio seguente; tuttavia, ignorerà l'affermazione. AWS STS

```
{
  "issuer": "https://example-domain.com",
  "jwks_uri": "https://example-domain.com/jwks/keys",
  "claims_supported": [
    "aud",
    "iat",
    "iss",
    "name",
    "sub",
    "custom"
  ],
  "response_types_supported": [
    "id_token"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "subject_types_supported": [
```

```
    "public"
  ]
}
```

Creazione e gestione di un provider OIDC (Console)

Segui queste istruzioni per creare e gestire un provider di identità OIDC IAM nella AWS Management Console.

> **Important**
>
> Se utilizzi un provider di identità OIDC di Google, Facebook o Amazon Cognito, non creare un provider di identità IAM separato utilizzando questa procedura. Questi provider di identità OIDC sono già integrati AWS e sono disponibili per l'uso da parte dell'utente. Segui invece i passaggi per creare nuovi ruoli per il provider di identità e consulta Creare un ruolo per la federazione OpenID Connect (console).

Come creare un provider di identità OIDC IAM (console)

1. Prima di creare un provider di identità OIDC IAM, occorre registrare l'applicazione sul provider di identità per ricevere un ID client. L'ID client (noto anche come destinatario) è un identificatore univoco per l'app rilasciato durante la registrazione dell'app sul provider di identità. Per ulteriori informazioni su come ottenere un ID client, consulta la documentazione per l'IdP.

   > **Note**
   >
   > AWS protegge la comunicazione con alcuni provider di identità OIDC (IdPs) tramite la nostra libreria di autorità di certificazione root (CA) affidabili anziché utilizzare l'impronta personale del certificato per verificare il certificato del server IdP. In questi casi, l'identificazione personale legacy rimane nella configurazione, ma non viene più utilizzata per la convalida. Questi OIDC IdPs includono Auth0, GitHub GitLab, Google e quelli che utilizzano un bucket Amazon S3 per ospitare un endpoint JSON Web Key Set (JWKS).

2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, scegli Provider di identità, quindi seleziona Aggiungi provider.

4. Per Configura provider, scegli OpenID Connect.
5. In Provider URL (URL provider), digitare l'URL del provider di identità. L'URL deve soddisfare queste restrizioni:
   - L'URL rileva la distinzione tra lettere maiuscole e minuscole.
   - L'URL deve iniziare con **`https://`**.
   - L'URL non deve contenere un numero di porta.
   - All'interno del tuo Account AWS, ogni provider di identità IAM OIDC deve utilizzare un URL univoco. Se provi a inviare un URL che è già stato utilizzato per un provider OpenID Connect in Account AWS, riceverai un errore.
6. Per Audience, digita l'ID client dell'applicazione che hai registrato con l'IdP e in cui hai ricevuto e a Step 1 cui hai inviato le richieste. AWS Se si dispone di ID client aggiuntivi (noti anche come destinatari) per questo provider di identità, è possibile aggiungerli in un secondo momento nella pagina dei dettagli del provider.

   > **Note**
   >
   > Se il tuo token IdP JWT include il `azp` claim, inserisci questo valore come valore Audience.

7. (Facoltativo) Per Aggiungi tag, puoi aggiungere coppie chiave-valore per aiutarti a identificare e organizzare i tuoi. IdPs È inoltre possibile utilizzare i tag per controllare l'accesso alle risorse AWS . Per ulteriori informazioni sul tagging dei provider di identità OIDC IAM, consulta Applicazione di tag ai provider di identità OpenID Connect (OIDC). Selezionare Aggiungi tag. Immetti i valori per ogni coppia chiave-valore del tag.
8. Controlla le informazioni inserite. Quando hai finito, scegli Aggiungi provider. IAM tenterà di recuperare e utilizzare l'impronta digitale della CA intermedia superiore del certificato del server IdP OIDC per creare il provider di identità IAM OIDC.

   > **Note**
   >
   > La catena di certificati del provider di identità OIDC deve iniziare con l'URL del dominio o dell'emittente, quindi con il certificato intermedio e terminare con il certificato radice. Se l'ordine della catena di certificati è diverso o include certificati duplicati o aggiuntivi, viene visualizzato un errore di mancata corrispondenza della firma e STS non riesce a convalidare il JSON Web Token (JWT). Correggi l'ordine dei certificati nella catena

restituita dal server per risolvere l'errore. Per ulteriori informazioni sugli standard della catena di certificati, consulta certificate_list nella RFC 5246 sul sito Web della serie RFC.

9. Assegna un ruolo IAM al tuo provider di identità per concedere alle identità degli utenti esterni gestite dal tuo provider di identità le autorizzazioni per accedere alle risorse del tuo account. AWS Per ulteriori informazioni sulla creazione di ruoli per la federazione delle identità, consulta Creazione di un ruolo per un provider di identità di terza parte (federazione).

> **Note**
>
> L'OIDC IdPs utilizzato in una politica di fiducia dei ruoli deve appartenere allo stesso account del ruolo che la considera attendibile.

Come aggiungere o rimuovere un'identificazione personale per un provider di identità OIDC IAM (console)

> **Note**
>
> AWS protegge la comunicazione con alcuni provider di identità OIDC (IdPs) tramite la nostra libreria di autorità di certificazione root (CA) affidabili anziché utilizzare l'impronta personale del certificato per verificare il certificato del server IdP. In questi casi, l'identificazione personale legacy rimane nella configurazione, ma non viene più utilizzata per la convalida. Questi OIDC IdPs includono Auth0, GitHub GitLab, Google e quelli che utilizzano un bucket Amazon S3 per ospitare un endpoint JSON Web Key Set (JWKS).

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità). Scegli quindi il nome del provider di identità IAM che desideri aggiornare.
3. Nella sezione Identificazioni personali, scegli Gestisci. Per immettere un nuovo valore di identificazione personale, scegli Aggiungi identificazione personale. Per rimuovere un'identificazione personale, scegli Rimuovi accanto all'elemento che desideri rimuovere.

> **Note**
>
> Un provider di identità OIDC IAM deve avere un numero di identificazioni personale compreso tra 1 e 5.

Al termine, scegli Salva modifiche.

Come aggiungere un destinatario per un provider di identità OIDC IAM (console)

1. Nel pannello di navigazione, scegli Provider di identità, quindi scegli il nome del provider di identità IAM che desideri aggiornare.
2. Nella sezione Destinatari, scegli Operazioni e seleziona Aggiungi destinatario.
3. Digita l'ID client dell'applicazione che hai registrato con l'IdP e in Step 1 cui hai ricevuto le richieste. AWS Quindi scegli Aggiungi destinatari.

> **Note**
>
> Un provider di identità OIDC IAM deve avere un numero di audience compreso tra 1 e 100.

Come rimuovere un destinatario da un provider di identità OIDC IAM (console)

1. Nel pannello di navigazione, scegli Provider di identità, quindi scegli il nome del provider di identità IAM che desideri aggiornare.
2. Nella sezione Destinatari, seleziona il pulsante di opzione accanto al destinatario che desideri rimuovere, quindi seleziona Operazioni.
3. Scegli Rimuovi destinatario. Viene visualizzata una nuova finestra.
4. Se rimuovi un destinatario, le identità a esso federate non possono assumere ruoli associati al destinatario. Nella finestra, leggi l'avviso e conferma di volere rimuovere il destinatario digitando la parola `remove` nel campo.
5. Scegli Rimuovi per rimuovere il destinatario.

Come eliminare un provider di identità OIDC IAM (console)

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).
3. Seleziona la casella di controllo accanto al provider di identità IAM che desideri eliminare. Viene visualizzata una nuova finestra.
4. Conferma che desideri eliminare il provider digitando la parola `delete` nel campo. Quindi, scegli Elimina.

Creazione e gestione di un provider di identità OIDC IAM (AWS CLI)

Puoi utilizzare i seguenti AWS CLI comandi per creare e gestire i provider di identità IAM OIDC.

Come creare un provider di identità OIDC IAM (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. Per creare un nuovo provider di identità OIDC IAM, esegui il comando:
   - `aws iam create-open-id-connect-provider`

Come aggiornare l'elenco di identificazioni personali del certificato del server per un provider di identità OIDC IAM esistente (AWS CLI)

- Per aggiornare l'elenco di identificazioni personali del certificato del server per un provider di identità OIDC IAM, esegui il comando:
  - `aws iam update-open-id-connect-provider-thumbprint`

Come aggiungere i tag a un provider di identità OIDC IAM esistente (AWS CLI)

- Per aggiungere i tag a un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam tag-open-id-connect-provider`

Come elencare i tag per un provider di identità OIDC IAM esistente (AWS CLI)

- Per elencare i tag per un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam list-open-id-connect-provider-tags`

Come rimuovere i tag da un provider di identità OIDC IAM (AWS CLI)

- Per rimuovere i tag da un provider di identità OIDC IAM esistente, esegui il comando:
  - `aws iam untag-open-id-connect-provider`

Come aggiungere o rimuovere un ID client da un provider di identità OIDC IAM esistente (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, esegui il comando:
   - `aws iam get-open-id-connect-provider`
3. Per aggiungere un nuovo ID client a un provider di identità OIDC IAM esistente, esegui il comando:
   - `aws iam add-client-id-to-open-id-connect-provider`
4. Per rimuovere un client da un provider di identità OIDC IAM esistente, esegui il comando:
   - `aws iam remove-client-id-from-open-id-connect-provider`

Come eliminare un provider di identità OIDC IAM (AWS CLI)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , emetti il seguente comando:
   - `aws iam list-open-id-connect-providers`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, esegui il comando:

- `aws iam get-open-id-connect-provider`

3. Per eliminare un provider di identità OIDC IAM, esegui il comando:

   - `aws iam delete-open-id-connect-provider`

Creazione e gestione di un provider di identità (API) OIDC AWS

Puoi utilizzare i seguenti comandi dell'API IAM per creare e gestire provider OIDC.

Per creare un provider di identità (API) IAM OIDC AWS

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:

   - `ListOpenIDConnectProviders`

2. Per creare un nuovo provider di identità OIDC IAM, chiama la seguente operazione:

   - `CreateOpenIDConnectProvider`

Per aggiornare l'elenco delle impronte digitali dei certificati server per un provider di identità (API) IAM OIDC esistente AWS

- Per aggiornare l'elenco di identificazioni personali del certificato del server per un provider di identità OIDC IAM, chiama la seguente operazione:

  - `UpdateOpenIDConnectProviderThumbprint`

Per etichettare un provider di identità (API) IAM OIDC esistente AWS

- Per aggiungere i tag a un provider di identità OIDC IAM, richiama la seguente operazione:

  - `TagOpenIDConnectProvider`

Per elencare i tag per un provider di identità (API) IAM OIDC esistente AWS

- Per elencare i tag per un provider di identità OIDC IAM esistente, richiama la seguente operazione:

- `ListOpenIDConnectProviderTags`

Per rimuovere i tag su un provider di identità (API) IAM OIDC esistente AWS

- Per rimuovere i tag da un provider di identità OIDC IAM esistente, richiama la seguente operazione:
  - `UntagOpenIDConnectProvider`

Come aggiungere o rimuovere un ID client da un provider di identità OIDC IAM esistente (API AWS)

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:
   - `ListOpenIDConnectProviders`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, chiama la seguente operazione:
   - `GetOpenIDConnectProvider`
3. Per aggiungere un nuovo ID client a un provider di identità OIDC IAM esistente, chiama la seguente operazione:
   - `AddClientIDToOpenIDConnectProvider`
4. Per rimuovere un ID client da un provider di identità OIDC IAM esistente, chiama la seguente operazione:
   - `RemoveClientIDFromOpenIDConnectProvider`

Per eliminare un provider di identità (API) IAM OIDC AWS

1. (Facoltativo) Per ottenere un elenco di tutti i provider di identità OIDC IAM nell'account AWS , chiama la seguente operazione:
   - `ListOpenIDConnectProviders`
2. (Facoltativo) Per ottenere informazioni dettagliate su un provider di identità OIDC IAM, chiama la seguente operazione:

- `GetOpenIDConnectProvider`

3. Per eliminare un provider di identità OIDC IAM, chiama la seguente operazione:

   - `DeleteOpenIDConnectProvider`

Ottieni l'impronta personale di un provider di identità OpenID Connect

Quando crei un provider di identità OpenID Connect (OIDC) in IAM, IAM richiede l'impronta digitale dell'autorità di certificazione (CA) intermedia superiore che ha firmato il certificato utilizzato dal provider di identità esterno (IdP). L'identificazione personale è una firma per il certificato della CA che è stato utilizzato per emettere il certificato per il provider di identità compatibile con OIDC. Quando crei un provider di identità IAM OIDC, ti fidi delle identità autenticate da quell'IdP per avere accesso al tuo. Account AWS Utilizzando l'impronta digitale del certificato della CA, ti fidi di qualsiasi certificato emesso da quella CA con lo stesso nome DNS di quello registrato. Questo elimina la necessità di aggiornare i trust in ogni account quando si rinnova il certificato di firma del provider di identità.

> **Important**
>
> Nella maggior parte dei casi, il server di federazione utilizza due certificati diversi:
>
> - Il primo stabilisce una connessione HTTPS tra AWS e il tuo IdP. Questo dovrebbe essere rilasciato da una CA root pubblica nota, ad esempio. AWS Certificate Manager Ciò consente al cliente di verificare l'affidabilità e lo stato del certificato.
> - Il secondo è usato per crittografare i token e deve essere firmato da un CA radice privato o pubblico.

Puoi creare un provider di identità IAM OIDC con AWS Command Line Interface gli strumenti per Windows o l' PowerShellAPI IAM. Quando utilizzi questi metodi, hai la possibilità di fornire manualmente un'impronta digitale. Se scegli di non includere un'impronta personale, IAM recupererà l'impronta personale CA intermedia superiore del certificato del server IdP OIDC. Se scegli di includere un'impronta digitale, devi ottenerla manualmente e fornirla a. AWS

Quando crei un provider di identità OIDC con la console IAM, IAM tenta di recuperare per te l'impronta personale della CA intermedia superiore del certificato del server IdP OIDC.

Ti consigliamo di ottenere anche manualmente l'impronta del tuo IdP OIDC e di verificare che IAM abbia recuperato l'impronta digitale corretta. Per ulteriori informazioni su come ottenere le impronte digitali dei certificati, consulta le seguenti sezioni.

> **Note**
>
> AWS protegge la comunicazione con alcuni provider di identità OIDC (IdPs) tramite la nostra libreria di autorità di certificazione root (CA) affidabili anziché utilizzare l'impronta personale del certificato per verificare il certificato del server IdP. In questi casi, l'identificazione personale legacy rimane nella configurazione, ma non viene più utilizzata per la convalida. Questi OIDC IdPs includono Auth0, GitHub GitLab, Google e quelli che utilizzano un bucket Amazon S3 per ospitare un endpoint JSON Web Key Set (JWKS).

Per risolvere i problemi più comuni relativi alla federazione IAM OIDC, consulta Risolvere gli errori relativi a OIDC su re:POST. AWS

Ottieni l'impronta digitale del certificato

Si utilizza un browser Web e lo strumento da riga di comando OpenSSL per ottenere l'impronta personale del certificato per un provider OIDC. Tuttavia, non è necessario ottenere manualmente l'impronta personale del certificato per creare un provider di identità IAM OIDC. È possibile utilizzare la procedura seguente per ottenere l'impronta personale del certificato del provider OIDC.

Per ottenere l'identificazione personale per un provider di identità OIDC

1. Prima di poter ottenere l'identificazione personale per un provider di identità OIDC, è necessario ottenere lo strumento a riga di comando OpenSSL. È possibile utilizzare questo strumento per scaricare la catena di certificati del provider di identità OIDC e produrre un'identificazione personale del certificato finale nella catena di certificati. Se è necessario installare e configurare OpenSSL, seguire le istruzioni in Installare OpenSSL e Configurare OpenSSL.
2. Inizia con l'URL del provider di identità OIDC (ad esempio, `https://server.example.com`) e quindi aggiungi `/.well-known/openid-configuration` per formare l'URL per il documento di configurazione del provider di identità, nel modo seguente:

   `https://`*`server.example.com`*`/.well-known/openid-configuration`

   Apri questo URL in un browser Web sostituendo *`server.example.com`* con il nome del server del provider di identità.

3. Nel documento visualizzato, utilizza l'opzione Trova del browser Web per individuare il testo `"jwks_uri"`. Subito dopo il testo `"jwks_uri"` sono presenti due punti (:) seguiti da un URL. Copiare il nome di dominio completo dell'URL. Non includere il percorso `https://` o qualsiasi altro percorso dopo il dominio di primo livello.

```
{
 "issuer": "https://accounts.example.com",
 "authorization_endpoint": "https://accounts.example.com/o/oauth2/v2/auth",
 "device_authorization_endpoint": "https://oauth2.exampleapis.com/device/code",
 "token_endpoint": "https://oauth2.exampleapis.com/token",
 "userinfo_endpoint": "https://openidconnect.exampleapis.com/v1/userinfo",
 "revocation_endpoint": "https://oauth2.exampleapis.com/revoke",
 "jwks_uri": "https://www.exampleapis.com/oauth2/v3/certs",
...
```

4. Utilizzare lo strumento a riga di comando OpenSSL per eseguire il seguente comando. Sostituire *keys.example.com* con il nome di dominio ottenuto in Step 3.

```
openssl s_client -servername keys.example.com -showcerts -
connect keys.example.com:443
```

5. Nella finestra di comando, scorrere verso l'alto fino a visualizzare un certificato simile al seguente esempio. Se viene visualizzato più di un certificato, individua l'ultimo certificato visualizzato (nella parte inferiore dell'output di comando). Contiene il certificato della migliore CA intermedia nella catena della certification authority.

```
-----BEGIN CERTIFICATE-----
 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
 YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
 21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
 rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
 nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
 NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
```

```
-----END CERTIFICATE-----
```

Copiare il certificato (incluse le righe `-----BEGIN CERTIFICATE-----` e `-----END CERTIFICATE-----`) e incollarlo in un file di testo. Salvare quindi il file con il nome **`certificate.crt`**.

> **Note**
>
> La catena di certificati del provider di identità OIDC deve iniziare con l'URL del dominio o dell'emittente, quindi con il certificato intermedio e terminare con il certificato radice. Se l'ordine della catena di certificati è diverso o include certificati duplicati o aggiuntivi, viene visualizzato un errore di mancata corrispondenza della firma e STS non riesce a convalidare il JSON Web Token (JWT). Correggi l'ordine dei certificati nella catena restituita dal server per risolvere l'errore. Per ulteriori informazioni sugli standard della catena di certificati, consulta certificate_list nella RFC 5246 sul sito Web della serie RFC.

6. Utilizzare lo strumento a riga di comando OpenSSL per eseguire il seguente comando.

```
openssl x509 -in certificate.crt -fingerprint -sha1 -noout
```

La finestra di comando visualizza l'identificazione personale del certificato, simile a quella dell'esempio seguente:

```
SHA1 Fingerprint=99:0F:41:93:97:2F:2B:EC:F1:2D:DE:DA:52:37:F9:C9:52:F2:0D:9E
```

Rimuovere i due punti (:) da questa stringa per ottenere l'identificazione personale finale, come segue:

```
990F4193972F2BECF12DDEDA5237F9C952F20D9E
```

7. Se stai creando il provider di identità IAM OIDC con Tools for Windows o l'API IAM AWS CLI, fornire un' PowerShellimpronta personale è facoltativo. Se scegli di non includere un'impronta personale durante la creazione, IAM recupererà l'impronta personale CA intermedia superiore del certificato del server IdP OIDC. Dopo aver creato il provider di identità IAM OIDC, puoi confrontare questa impronta personale con quella recuperata da IAM.

   Se stai creando il provider di identità IAM OIDC nella console IAM, la console tenta di recuperare per te l'impronta personale della CA intermedia superiore del certificato del server IdP OIDC.

Puoi confrontare questa impronta personale con quella recuperata da IAM. Dopo aver creato il provider di identità IAM OIDC, puoi visualizzare l'impronta personale del provider di identità IAM OIDC nella scheda di verifica degli endpoint nella pagina di riepilogo della console di riepilogo del provider OIDC.

> **Important**
>
> Se l'impronta personale che hai ottenuto non corrisponde a quella che vedi nei dettagli dell'impronta personale del provider di identità IAM OIDC, non dovresti utilizzare il provider OIDC. Invece, dovresti eliminare il provider OIDC creato e riprovare a creare il provider OIDC dopo un certo periodo di tempo. Verifica che le impronte digitali corrispondano prima di utilizzare il provider. Se dopo un secondo tentativo non vi è comunque corrispondenza tra le identificazioni personali, accedere al forum di IAM per contattare AWS.

Installare OpenSSL

Se OpenSSL non è già installato, segui le istruzioni in questa sezione.

Come installare OpenSSL su Linux e Unix

1. Passa a OpenSSL: Source, Tarballs (https://openssl.org/source/).
2. Scarica l'origine più recente e compila il pacchetto.

Per installare OpenSSL in ambiente Windows

1. Vai a OpenSSL: Distribuzioni binarie (https://wiki.openssl.org/index.php/Binaries) per un elenco di siti da cui è possibile installare la versione di Windows.
2. Segui le istruzioni sul sito selezionato per avviare l'installazione.
3. Se viene richiesto di installare Microsoft Visual C++ 2008 Redistributables e questo non è già installato sul tuo sistema, scegli il link di download appropriato per il tuo ambiente. Segui le istruzioni fornite dalla Installazione guidata di Microsoft Visual C++ 2008 Redistributable.

> **Note**
>
> Se non sei sicuro che Microsoft Visual C++ 2008 Redistributables sia già installato nel sistema, puoi provare a installare prima OpenSSL. Il programma di installazione di

OpenSSL visualizza un avviso se Microsoft Visual C++ 2008 Redistributables non è ancora installato. Assicurati di installare l'architettura (32 bit o 64 bit) che corrisponde alla versione di OpenSSL installata.

4. Dopo aver installato Microsoft Visual C++ 2008 Redistributables, seleziona la versione appropriata dei file binari OpenSSL per l'ambiente e salva il file in locale. Avvia l'Installazione guidata di OpenSSL.
5. Segui le istruzioni descritte in Installazione guidata di OpenSSL.

Configurare OpenSSL

Prima di utilizzare i comandi OpenSSL, è necessario configurare il sistema operativo in modo che contenga le informazioni sulla posizione in cui è installato OpenSSL.

Come configurare OpenSSL su Linux o Unix

1. Alla riga di comando, imposta la variabile `OpenSSL_HOME` sulla posizione dell'installazione di OpenSSL:

```
$ export OpenSSL_HOME=path_to_your_OpenSSL_installation
```

2. Configura il percorso in modo da includere l'installazione di OpenSSL:

```
$ export PATH=$PATH:$OpenSSL_HOME/bin
```

> **Note**
>
> Eventuali modifiche apportate alle variabili di ambiente con il comando `export` sono valide solo per la sessione corrente. Puoi apportare modifiche permanenti alle variabili di ambiente impostandole nel file di configurazione della shell. Per ulteriori informazioni, consulta la documentazione relativa al sistema operativo in uso.

Come configurare OpenSSL su Windows

1. Apri una finestra del prompt dei comandi.
2. Configura la variabile `OpenSSL_HOME` sulla posizione dell'installazione di OpenSSL:

```
C:\> set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

3. Imposta la variabile `OpenSSL_CONF` sulla posizione del file di configurazione nell'installazione di OpenSSL:

```
C:\> set OpenSSL_CONF=path_to_your_OpenSSL_installation\bin\openssl.cfg
```

4. Configura il percorso in modo da includere l'installazione di OpenSSL:

```
C:\> set Path=%Path%;%OpenSSL_HOME%\bin
```

> **Note**
>
> Eventuali modifiche apportate alle variabili di ambiente Windows in una finestra del prompt dei comandi sono valide solo per la sessione della riga di comando corrente. È possibile apportare modifiche permanenti alle variabili di ambiente impostandole come proprietà di sistema. Le procedure esatte dipendono dalla versione di Windows in uso. (Ad esempio, su Windows 7, apri Pannello di controllo, Sistema e sicurezza, Sistema. Quindi scegli Impostazioni di sistema avanzate, scheda Avanzate, Variabili di ambiente.) Per ulteriori informazioni, consulta la documentazione di Windows.

Risorse aggiuntive per la federazione OIDC

Le seguenti risorse possono aiutarti a saperne di più sulla federazione OIDC:

- Usa OpenID Connect all'interno dei tuoi GitHub flussi di lavoro configurando OpenID Connect in Amazon Web Services
- Amazon Cognito Identity nella Guida alle librerie Amplify per Android e Guida all'identità di Amazon Cognito nella Guida alle librerie Amplify per Swift.
- Automating IAM Web Identity Roles con Microsoft Entra ID on AWS the Partner Network (APN) AWS basati su OpenID Connect spiega come autenticare processi o applicazioni automatizzati in background eseguiti al di fuori dell'autorizzazione OIDC. AWS machine-to-machine
- L'articolo Web Identity Federation with Mobile Applications descrive la federazione OIDC e mostra un esempio di come utilizzare la federazione OIDC per accedere ai contenuti in Amazon S3.

Federazione SAML 2.0

AWS supporta la federazione delle identità con SAML 2.0 (Security Assertion Markup Language 2.0), uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere AWS Management Console o chiamare le operazioni AWS API senza che tu debba creare un utente IAM per tutti i membri dell'organizzazione. Utilizzando SAML, puoi semplificare il processo di configurazione della federazione AWS, poiché puoi utilizzare il servizio dell'IdP invece di scrivere un codice proxy di identità personalizzato.

La federazione IAM supporta i casi d'uso indicati di seguito.

- Accesso federato per consentire a un utente o a un'applicazione dell'organizzazione di chiamare le operazioni AWS API. Questo caso d'uso è discusso nella sezione seguente. viene utilizzata un'asserzione SAML (come parte della risposta di autenticazione) generata nell'organizzazione per ottenere credenziali di sicurezza temporanee. Questo scenario è analogo ad altri scenari di federazione supportati da IAM, come descritto in Richiesta di credenziali di sicurezza temporanee e Federazione OIDC. Tuttavia, l'organizzazione basata su SAML 2.0 gestisce molti dettagli IdPs in fase di esecuzione per eseguire il controllo dell'autenticazione e delle autorizzazioni.
- Single Sign-On (SSO) basato sul Web da e verso l'organizzazione. AWS Management Console Gli utenti possono accedere a un portale dell'organizzazione ospitato da un IdP compatibile con SAML 2.0, selezionare un'opzione a cui accedere ed essere reindirizzati AWS alla console senza dover fornire ulteriori informazioni di accesso. Puoi utilizzare un IdP SAML di terze parti per stabilire l'accesso SSO alla console o creare un IdP personalizzato per consentire l'accesso alla console per utenti esterni. Per ulteriori informazioni sulla creazione di un provider di identità personalizzato, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

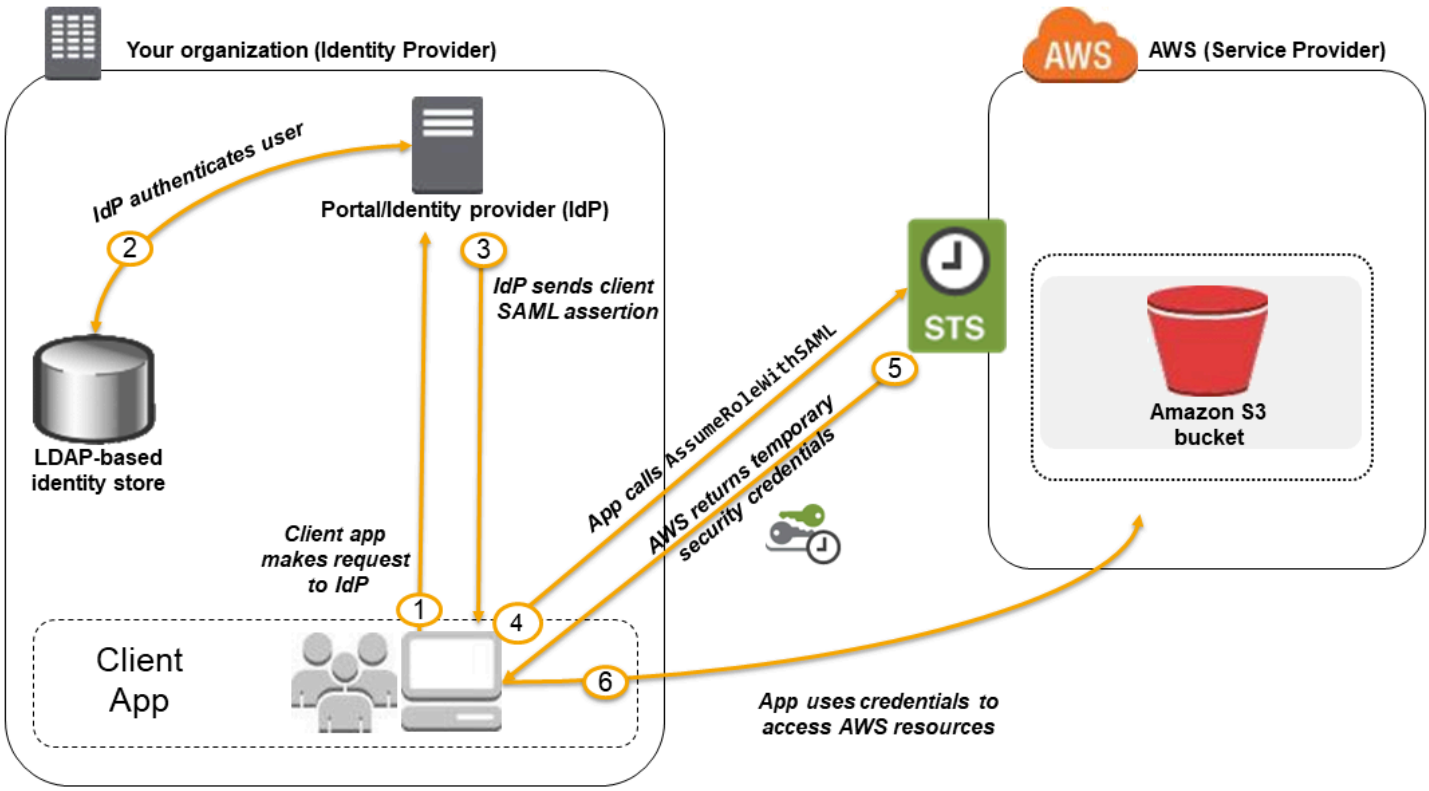
Argomenti

- Utilizzo della federazione basata su SAML per l'accesso API ad AWS
- Panoramica della configurazione della federazione basata su SAML 2.0
- Panoramica del ruolo per consentire l'accesso federato SAML alle tue risorse AWS
- Identificazione univoca degli utenti nella federazione basata su SAML
- Crea un provider di identità SAML in IAM
- Configura il tuo IdP SAML 2.0 con la fiducia dei relying party e l'aggiunta di claim
- Integra fornitori di soluzioni SAML di terze parti con AWS

- Configurare le asserzioni SAML per la risposta di autenticazione
- Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console

Utilizzo della federazione basata su SAML per l'accesso API ad AWS

Supponi di voler fornire ai dipendenti un modo per copiare i dati dai loro computer a una cartella di backup. È possibile creare un'applicazione che gli utenti possono eseguire sui propri computer. Sul back-end, l'applicazione legge e scrive oggetti in un bucket Amazon S3. Gli utenti non hanno accesso diretto a. AWS viene utilizzato invece il seguente processo:



1. Un utente dell'organizzazione utilizza un'app client per richiedere l'autenticazione dal provider di identità della propria organizzazione.
2. Il provider di identità autentica l'utente rispetto all'archivio identità organizzazione.
3. Il provider di identità crea un'asserzione SAML con informazioni sull'utente e invia l'asserzione all'app client.
4. L'app client chiama l' AWS STS `AssumeRoleWithSAML`API, passando l'ARN del provider SAML, l'ARN del ruolo da assumere e l'asserzione SAML di IdP.
5. La risposta dell'API all'app client include credenziali di sicurezza temporanee.

6. L'app client utilizza le credenziali di sicurezza temporanee per chiamare le operazioni dell'API di Amazon S3.

Panoramica della configurazione della federazione basata su SAML 2.0

Prima di poter utilizzare la federazione basata su SAML 2.0 come descritto nello scenario e nel diagramma precedenti, devi configurare l'IdP dell'organizzazione e il tuo in modo che si fidino l'uno dell'altro. Account AWS Di seguito è descritta la procedura generale per la configurazione di questo tipo di attendibilità. All'interno dell'organizzazione, è necessario disporre di un provider di identità che supporti SAML 2.0, come Microsoft Active Directory Federation Service (ADFS, parte di Windows Server), Shibboleth o di un altro provider compatibile con SAML 2.0.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per i dettagli, consulta l'articolo del AWS Security Blog Come utilizzare gli endpoint SAML regionali per il failover.

Configura l'IdP della tua organizzazione e fidati AWS l'uno dell'altro

1. Registrati AWS come fornitore di servizi (SP) con l'IdP della tua organizzazione. Utilizzo del documento dei metadati SAML generato da `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`

   Per un elenco dei possibili valori di *region-code*, consulta la colonna Region (Regione) in Endpoint di accesso AWS.

   Se lo desideri, puoi utilizzare il documento dei metadati SAML generati da `https://signin.aws.amazon.com/static/saml-metadata.xml` .
2. Utilizzando il provider di identità dell'organizzazione, generare un file XML di metadati equivalenti che possono descrivere l'IdP come provider di identità IAM in AWS. Deve includere il nome dell'emittente, una data di creazione, una data di scadenza e le chiavi che AWS possono essere utilizzate per convalidare le risposte di autenticazione (asserzioni) dell'organizzazione.
3. Nella console IAM, crei un provider di identità SAML. Come parte di questo processo, carichi il documento di metadati SAML prodotti dall'IdP della tua organizzazione in. Step 2 Per ulteriori informazioni, consulta Crea un provider di identità SAML in IAM.

4. In IAM, vengono creati uno o più ruoli IAM. Nella politica di fiducia del ruolo, imposti il provider SAML come principale, che stabilisce una relazione di fiducia tra la tua organizzazione e. AWS La policy di autorizzazione del ruolo stabilisce le operazioni che gli utenti dell'organizzazione possono effettuare in AWS. Per ulteriori informazioni, consulta Creazione di un ruolo per un provider di identità di terza parte (federazione).

   > **Note**
   >
   > Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

5. Nel provider di identità dell'organizzazione, si definiscono asserzioni che associano utenti o gruppi dell'organizzazione ai ruoli IAM. Nota che i diversi utenti e gruppi dell'organizzazione potrebbero essere mappati a diversi ruoli IAM. La procedura per eseguire la mappatura dipende dal provider di identità che si sta utilizzando. Nello scenario precedente che utilizza una cartella Amazon S3 per gli utenti, è possibile che tutti gli utenti dispongano della mappatura allo stesso ruolo che fornisce le autorizzazioni Amazon S3. Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

   Se il tuo IdP abilita l'SSO AWS sulla console, puoi configurare la durata massima delle sessioni della console. Per ulteriori informazioni, consulta Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console.

6. Nell'applicazione che stai creando, chiami l' AWS Security Token Service `AssumeRoleWithSAML`API, passandole l'ARN del provider SAML in cui hai creato, Step 3 l'ARN del ruolo da assumere in cui hai creato e l'asserzione SAML sull'utente corrente che ricevi dal tuo IdP. Step 4 AWS si assicura che la richiesta di assunzione del ruolo provenga dall'IdP a cui si fa riferimento nel provider SAML.

   Per ulteriori informazioni, consulta AssumeRoleWithSAML nell'API Reference.AWS Security Token Service

7. Se la richiesta ha esito positivo, l'API restituisce una serie di credenziali di sicurezza temporanee, che l'applicazione può utilizzare per inviare richieste firmate ad AWS. L'applicazione contiene informazioni sull'utente corrente e può accedere a cartelle specifiche dell'utente in Amazon S3, come descritto nello scenario precedente.

Panoramica del ruolo per consentire l'accesso federato SAML alle tue risorse AWS

Il ruolo o i ruoli creati in IAM definiscono ciò che gli utenti federati dell'organizzazione possono fare in AWS. Quando si creano policy di affidabilità per il ruolo, è necessario specificare il provider SAML creato in precedenza come `Principal`. È inoltre possibile estendere la policy di affidabilità con un elemento `Condition` per permettere solo agli utenti che corrispondono ad alcuni attributi SAML di accedere al ruolo. Ad esempio, è possibile specificare che solo gli utenti con l'affiliazione SAML `staff` (come affermato da https://openidp.feide.no) possono accedere al ruolo, come illustrato dalla seguente policy di esempio:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"Federated": "arn:aws:iam::account-id:saml-provider/
ExampleOrgSSOProvider"},
    "Action": "sts:AssumeRoleWithSAML",
    "Condition": {
      "StringEquals": {
        "saml:aud": "https://signin.aws.amazon.com/saml",
        "saml:iss": "https://openidp.feide.no"
      },
      "ForAllValues:StringLike": {"saml:edupersonaffiliation": ["staff"]}
    }
  }]
}
```

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

Per ulteriori informazioni sulle chiavi SAML che è possibile verificare in una policy, consulta Chiavi disponibili per la federazione AWS STS basata su SAML.

Puoi includere endpoint regionali per l'attributo `saml:aud` in `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`. Per un elenco dei possibili valori di *region-code*, consulta la colonna Region (Regione) in Endpoint di accesso AWS.

Per la policy di autorizzazione nel ruolo, è necessario specificare le autorizzazioni come per qualsiasi ruolo. Ad esempio, se agli utenti della tua organizzazione è consentito amministrare istanze di Amazon Elastic Compute Cloud, devi consentire esplicitamente le azioni di Amazon EC2 nella politica delle autorizzazioni, come quelle nella politica gestita di Amazon EC2. FullAccess

Identificazione univoca degli utenti nella federazione basata su SAML

Quando si creano policy di accesso in IAM, spesso è utile poter specificare le autorizzazioni in base all'identità degli utenti. Ad esempio, per gli utenti che sono stati federati tramite SAML, un'applicazione potrebbe voler mantenere le informazioni in Amazon S3 utilizzando una struttura come questa:

```
myBucket/app1/user1
myBucket/app1/user2
myBucket/app1/user3
```

Puoi creare il bucket (`myBucket`) e la cartella (`app1`) tramite la console Amazon S3 o AWS CLI il, poiché si tratta di valori statici. Tuttavia, le cartelle specifiche dell'utente (*`user1`*, *`user2`*, *`user3`* e così via) devono essere create in fase di runtime utilizzando il codice, poiché il valore che identifica l'utente non è noto fino al primo accesso dell'utente tramite il processo di federazione.

Per scrivere policy che fanno riferimento ai dettagli specifici dell'utente come parte di un nome di risorsa, l'identità dell'utente deve essere disponibile nelle chiavi SAML che possono essere utilizzate nelle condizioni di policy. Le seguenti chiavi sono disponibili per la federazione basata su SAML 2.0 da utilizzare nelle policy IAM. È possibile utilizzare i valori restituiti dalle chiavi seguenti per creare identificativi utente univoci per risorse come le cartelle Amazon S3.

- `saml:namequalifier`. Un valore hash basato sulla concatenazione del valore della risposta `Issuer` (`saml:iss`) e una stringa con l'ID account `AWS` e il nome descrittivo (l'ultima parte dell'ARN) del provider SAML in IAM. La concatenazione dell'ID account e del nome descrittivo del provider SAML è disponibile per le policy IAM sotto forma di chiave `saml:doc`. L'ID account e il nome del provider devono essere separati da una barra "/" come in "123456789012/provider_name". Per ulteriori informazioni, consulta la chiave `saml:doc` in Chiavi disponibili per la federazione AWS STS basata su SAML.

  La combinazione di `NameQualifier` e `Subject` può essere utilizzata per identificare in modo univoco un utente federato. Il seguente pseudocodice mostra come viene calcolato questo valore. In questo pseudocodice + indica la concatenazione, `SHA1` rappresenta una funzione che produce

un digest del messaggio utilizzando SHA-1 e `Base64` rappresenta una funzione che genera una versione con codificazione Base-64 dell'output hash.

```
Base64 ( SHA1 ( "https://example.com/saml" + "123456789012" + "/
MySAMLIdP" ) )
```

Per ulteriori informazioni sulle chiavi di policy disponibili quando si utilizza la federazione basata su SAML, consulta Chiavi disponibili per la federazione AWS STS basata su SAML.

- `saml:sub` (Stringa). Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).
- `saml:sub_type` (Stringa). Questa chiave può essere `persistent`, `transient` o l'URI `Format` completo degli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Il valore `persistent` indica che il valore in `saml:sub` è lo stesso per un utente in tutte le sessioni. Se il valore è `transient`, l'utente dispone di un valore `saml:sub` diverso per ogni sessione. Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

L'esempio seguente mostra una policy di autorizzazione che utilizza le chiavi precedenti per concedere le autorizzazioni a una cartella specifica per utente in Amazon S3. La policy presuppone che gli oggetti Amazon S3 vengano identificati utilizzando un prefisso che include sia `saml:namequalifier` che `saml:sub`. Si noti che l'elemento `Condition` include un test per assicurarsi che `saml:sub_type` sia impostato su `persistent`. Se è impostato su `transient`, il valore `saml:sub` per l'utente può essere diverso per ogni sessione e la combinazione di valori non deve essere utilizzata per identificare le cartelle specifiche dell'utente.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": [
      "arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}",
      "arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}/*"
    ],
```

```
    "Condition": {"StringEquals": {"saml:sub_type": "persistent"}}
  }
}
```

Per ulteriori informazioni sulle asserzioni di mappatura dal provider di identità alle chiavi di policy, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

Crea un provider di identità SAML in IAM

Un provider di identità SAML 2.0 IAM è un'entità in IAM che descrive un servizio del provider di identità (IdP) esterno che supporta lo standard SAML 2.0 (Security Assertion Markup Language 2.0). Utilizzi un provider di identità IAM quando desideri stabilire un rapporto di fiducia tra un IdP compatibile con SAML come Shibboleth o Active Directory Federation Services e AWS consentire agli utenti dell'organizzazione di accedere alle risorse. AWS I provider di identità SAML IAM vengono utilizzati come principali nelle policy di attendibilità IAM.

Per ulteriori informazioni su questo scenario, consulta Federazione SAML 2.0.

Puoi creare e gestire un provider di identità IAM nelle AWS Management Console o con AWS CLI, Tools for Windows o chiamate API. PowerShell AWS

Una volta creato un provider SAML, dovrai creare uno o più ruoli IAM. Un ruolo è un'identità AWS che non ha le proprie credenziali (come invece fa un utente). Tuttavia, in questo contesto, un ruolo viene assegnato in modo dinamico a un utente federato autenticato dall'IdP dell'organizzazione. Il ruolo consente al provider di identità dell'organizzazione di richiedere credenziali di sicurezza provvisorie per l'accesso a AWS. Le politiche assegnate al ruolo determinano le operazioni consentite agli utenti federati. AWS Per creare un ruolo per una federazione SAML consultare Creazione di un ruolo per un provider di identità di terza parte (federazione).

Infine, dopo aver creato il ruolo, completi il trust SAML configurando il tuo IdP con le informazioni AWS e i ruoli che desideri vengano utilizzati dagli utenti federati. Questa operazione viene definita configurazione di una relazione di trust fra IdP e AWS. Per configurare una relazione di trust, consultare Configura il tuo IdP SAML 2.0 con la fiducia dei relying party e l'aggiunta di claim.

Argomenti

- Prerequisiti
- Crea e gestisci un provider di identità IAM SAML (console)
- Crea e gestisci un IAM SAML Identity Provider ()AWS CLI

- Crea e gestisci un provider di identità IAM SAML (API)AWS

Prerequisiti

Prima di poter creare un provider di identità SAML, devi avere le seguenti informazioni dal tuo IdP.

- Ottieni il documento di metadati SAML dal tuo IdP. Questo documento include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) che vengono ricevute dal provider di identità. Per generare il documento di metadati, utilizza il software di gestione delle identità fornito dal tuo IdP esterno.

  > **Important**
  >
  > Questo file di metadati include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) ricevute dal provider di identità. Il file di metadati deve essere codificati in formato UTF-8, senza BOM (Byte Order Mark). Per rimuovere il BOM, codifica i file come UTF-8 utilizzando un editor di testi come ad esempio Notepad++.
  > Il certificato x.509 incluso come parte del documento di metadati SAML deve utilizzare una chiave di almeno 1024 bit. Inoltre, il certificato x.509 deve essere privo di eventuali estensioni ripetute. È possibile utilizzare le estensioni, ma possono essere visualizzate una sola volta nel certificato. Se il certificato x.509 non soddisfa nessuna delle due condizioni, la creazione dell'IdP ha esito negativo e restituisce l'errore "Unable to parse metadata". Come definito dal profilo di interoperabilità dei metadati SAML V2.0 versione 1.0, IAM non valuta né interviene in merito alla scadenza del certificato X.509 del documento di metadati.

Per istruzioni su come configurare molti dei file disponibili con cui IdPs lavorare AWS, incluso come generare il documento di metadati SAML richiesto, consulta. Integra fornitori di soluzioni SAML di terze parti con AWS

Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Crea e gestisci un provider di identità IAM SAML (console)

Puoi utilizzarlo AWS Management Console per creare, aggiornare ed eliminare i provider di identità IAM SAML. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Come creare un provider di identità SAML IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Provider di identità, quindi seleziona Aggiungi provider.
3. Per Configura provider, scegli SAML.
4. Digitare un nome per il provider di identità.
5. In Documento metadati, fai clic su Scegli file e specifica il documento di metadati SAML scaricato in the section called “Prerequisiti”.
6. (Facoltativo) Per Aggiungi tag puoi aggiungere coppie chiave-valore per aiutarti a identificare e organizzare le tue. IdPs È inoltre possibile utilizzare i tag per controllare l'accesso alle risorse AWS . Per ulteriori informazioni sull'applicazione di tag ai provider di identità SAML, vedere Tagging di provider di identità SAML per IAM.

   Selezionare Aggiungi tag. Immetti i valori per ogni coppia chiave-valore del tag.
7. Controlla le informazioni inserite. Quando hai finito, scegli Aggiungi provider.
8. Assegna un ruolo IAM al tuo provider di identità per concedere alle identità degli utenti esterni gestite dal tuo provider di identità le autorizzazioni per accedere AWS alle risorse del tuo account. Per ulteriori informazioni sulla creazione di ruoli per la federazione delle identità, consulta Creazione di un ruolo per un provider di identità di terza parte (federazione).

   > **Note**
   >
   > Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

Per eliminare un provider SAML (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Identity providers (Provider di identità).
3. Seleziona la casella di controllo accanto al provider di identità che desideri eliminare.
4. Scegli Elimina. Viene visualizzata una nuova finestra.
5. Conferma che desideri eliminare il provider digitando la parola `delete` nel campo. Quindi, scegli Elimina.

Crea e gestisci un IAM SAML Identity Provider ()AWS CLI

Puoi utilizzarlo AWS CLI per creare, aggiornare ed eliminare i provider SAML. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Come creare un provider di identità IAM e caricare un documento di metadati (AWS CLI)

- Eseguire il comando: `aws iam create-saml-provider`

Per aggiornare un provider di identità IAM SAML ()AWS CLI

- Eseguire il comando: `aws iam update-saml-provider`

Come aggiungere i tag a un provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam tag-saml-provider`

Come elencare i tag per il provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam list-saml-provider-tags`

Come rimuovere i tag da un provider di identità IAM esistente (AWS CLI)

- Eseguire il comando: `aws iam untag-saml-provider`

Come eliminare un provider di identità SAML IAM (AWS CLI)

1. (Facoltativo) Per elencare le informazioni di tutti i provider (ad esempio l'ARN, la data di creazione e la scadenza), eseguire il seguente comando:
   - `aws iam list-saml-providers`
2. (Facoltativo) Per ottenere informazioni su un provider specifico, ad esempio ARN, data di creazione, data di scadenza, impostazioni di crittografia e informazioni sulla chiave privata, esegui il comando seguente:
   - `aws iam get-saml-provider`
3. Per eliminare un provider di identità IAM, esegui il comando:

- `aws iam delete-saml-provider`

Crea e gestisci un provider di identità IAM SAML (API)AWS

Puoi utilizzare l' AWS API per creare, aggiornare ed eliminare i provider SAML. Per assistenza sulla federazione SAML, consulta Risoluzione dei problemi della federazione SAML.

Per creare un provider di identità IAM e caricare un documento di metadati (API)AWS

- Richiamare l'operazione: `CreateSAMLProvider`

Per aggiornare un provider di identità IAM SAML (API)AWS

- Richiamare l'operazione: `UpdateSAMLProvider`

Per etichettare un provider di identità (API) IAM esistente AWS

- Richiamare l'operazione: `TagSAMLProvider`

Per elencare i tag per un provider di identità IAM (AWS API) esistente

- Richiamare l'operazione: `ListSAMLProviderTags`

Per rimuovere i tag su un provider di identità IAM (AWS API) esistente

- Richiamare l'operazione: `UntagSAMLProvider`

Per eliminare un provider di identità IAM (AWS API)

1. (Facoltativo) Per elencare informazioni per tutti IdPs, come l'ARN, la data di creazione e la scadenza, chiamate la seguente operazione:
   - `ListSAMLProviders`
2. (Facoltativo) Per ottenere informazioni su un provider specifico, come ARN, data di creazione, data di scadenza, impostazioni di crittografia e informazioni sulla chiave privata, eseguire la seguente operazione:

- `GetSAMLProvider`

3. Per eliminare un IdP, richiamare la seguente operazione:

   - `DeleteSAMLProvider`

Configura il tuo IdP SAML 2.0 con la fiducia dei relying party e l'aggiunta di claim

Quando crei un provider di identità IAM e un ruolo per l'accesso SAML, indichi ad AWS il provider di identità (IdP) esterno e le operazioni che i rispettivi utenti sono autorizzati a effettuare. Il passaggio successivo consiste nell'informare l'IdP in AWS qualità di fornitore di servizi. Questa relazione è nota come relazione di trust tra il provider di identità e AWS. L'esatto processo per aggiungere una relazione di trust dipende dall'IdP utilizzato. Per ulteriori informazioni, consulta la documentazione relativa al tuo software di gestione delle identità.

Molti IdPs consentono di specificare un URL da cui l'IdP può leggere un documento XML contenente informazioni e certificati del relying party. Per AWS, usa `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml` o. `https://signin.aws.amazon.com/static/saml-metadata.xml` Per un elenco dei possibili valori di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS.

Se non è possibile specificare direttamente un URL, scaricare il documento XML dall'URL precedente e importarlo nel software dell'IdP.

Inoltre, devi creare regole di reclamo appropriate nel tuo IdP che specifichino AWS come parte affidataria. Quando l'IdP invia una risposta SAML all' AWS endpoint, include un'asserzione SAML che contiene una o più attestazioni. Un'attestazione consiste in un set di informazioni sull'utente e sui rispettivi gruppi. Una regola di attestazione mappa tali informazioni negli attributi SAML. Ciò ti consente di assicurarti che le risposte di autenticazione SAML del tuo IdP contengano gli attributi AWS necessari utilizzati nelle policy IAM per verificare le autorizzazioni per gli utenti federati. Per ulteriori informazioni, consulta i seguenti argomenti:

- Panoramica del ruolo per consentire l'accesso federato SAML alle tue risorse AWS. In questo argomento viene descritto l'uso di chiavi specifiche SAML nelle policy IAM e le modalità di utilizzo per limitare le autorizzazioni per gli utenti federati SAML.
- Configurare le asserzioni SAML per la risposta di autenticazione. In questo argomento viene descritto come configurare le attestazioni SAML che includono informazioni sull'utente. Le attestazioni sono raggruppate in un'asserzione SAML e incluse nella risposta SAML inviata ad

AWS. Devi assicurarti che le informazioni necessarie alle AWS policy siano incluse nell'asserzione SAML in un formato riconoscibile e utilizzabile. AWS

- Integra fornitori di soluzioni SAML di terze parti con AWS. Questo argomento fornisce collegamenti alla documentazione fornita da organizzazioni di terze parti su come integrare soluzioni di identità con AWS.

> **Note**
>
> Per migliorare la resilienza della federazione, ti consigliamo di configurare l'IdP e la federazione AWS per supportare più endpoint di accesso SAML. Per i dettagli, consulta l'articolo del AWS Security Blog Come utilizzare gli endpoint SAML regionali per il failover.

Integra fornitori di soluzioni SAML di terze parti con AWS

> **Note**
>
> Ti consigliamo di richiedere agli utenti umani di utilizzare credenziali temporanee per l'accesso. AWS Hai preso in considerazione l'idea di utilizzarlo AWS IAM Identity Center? Puoi utilizzare IAM Identity Center per gestire centralmente l'accesso a più account Account AWS e fornire agli utenti un accesso Single Sign-On protetto da MFA a tutti gli account assegnati da un'unica posizione. Con IAM Identity Center puoi creare e gestire le identità degli utenti in IAM Identity Center o connetterti facilmente al tuo attuale gestore dell'identità digitale (IdP) compatibile con SAML 2.0. Per ulteriori informazioni, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

I seguenti collegamenti consentono di configurare soluzioni di provider di identità (IdP) SAML 2.0 di terze parti in modo che AWS funzionino con la federazione.

> **Tip**
>
> AWS I tecnici dell'assistenza possono assistere i clienti che dispongono di piani di supporto aziendali e aziendali con alcune attività di integrazione che coinvolgono software di terze parti. Per un elenco aggiornato delle piattaforme e delle applicazioni supportate, consulta Qual è il software per terze parti supportato? in Domande frequenti sul supporto di AWS .

Soluzione	Ulteriori informazioni
Auth0	Integrazione con Amazon Web Services: questa pagina del sito Web di documentazione Auth0 contiene collegamenti a risorse che descrivono come configurare il Single Sign-On (SSO) con AWS Management Console e include un esempio. JavaScript È possibile configurare Auth0 per passare i tag di sessione. Per ulteriori informazioni, consulta Auth0 annuncia una partnership con i tag di sessione IAM. AWS
Microsoft Entra	Tutorial: integrazione SSO di Microsoft Entra con AWS Single-Account Access — Questo tutorial sul sito Web di Microsoft descrive come configurare Microsoft Entra (precedentemente noto come Azure AD) come provider di identità (IdP) utilizzando la federazione SAML.
Centrify	Configura Centrify e usa SAML per SSO AWS: questa pagina del sito Web di Centrify spiega come configurare Centrify per utilizzare SAML per SSO. AWS
CyberArk	Configura CyberArkper fornire l'accesso ad Amazon Web Services (AWS) agli utenti che accedono tramite SAML Single Sign-On (SSO) dal portale utenti. CyberArk
ForgeRock	La ForgeRock piattaforma Identity si integra con. AWS Puoi configurare ForgeRock per passare i tag di sessione. Per ulteriori informazioni, consulta Attribute Based Access Control for Amazon Web Services.
Google Workspace	Applicazione cloud Amazon Web Services: questo articolo sul sito di assistenza per amministratori di Google Workspace descrive come configurare Google Workspace come IdP SAML 2.0 e come fornitore di AWS servizi.
IBM	È possibile configurare IBM per passare i tag di sessione. Per ulteriori informazioni, consulta IBM Cloud Identity IDaaS, uno dei primi a supportare i tag di sessione. AWS

Soluzione	Ulteriori informazioni
JumpCloud	Concessione dell'accesso tramite IAM Roles for Single Sign On (SSO) con Amazon AWS: questo articolo sul JumpCloud sito Web descrive come configurare e abilitare l'SSO basato sui ruoli IAM per. AWS
Matrix42	MyWorkspace Guida introduttiva: questa guida descrive come integrare i servizi di AWS identità con Matrix42. MyWorkspace
Microsoft Active Directory Federation Services (AD FS)	Note sul campo: Integrazione di Active Directory Federation Service con AWS IAM Identity Center — Questo post sul blog di AWS architettura spiega il flusso di autenticazione tra AD FS e AWS IAM Identity Center (IAM Identity Center). IAM Identity Center supporta la federazione delle identità con SAML 2.0, consentendo l'integrazione con le soluzioni AD FS. Gli utenti possono accedere al portale di IAM Identity Center con le proprie credenziali aziendali, riducendo il sovraccarico amministrativo dovuto al mantenimento di credenziali separate. Inoltre, puoi configurare AD FS per passare i tag di sessione. Per ulteriori informazioni, consulta Utilizzo del controllo accessi basato sugli attributi con AD FS per semplificare la gestione delle autorizzazioni IAM.
miniOrange	SSO per AWS: questa pagina del sito Web MiniOrange descrive come stabilire un accesso sicuro AWS per le aziende e il pieno controllo sull'accesso alle AWS applicazioni.

Soluzione	Ulteriori informazioni
Okta	Integrazione dell'interfaccia a riga di comando di Amazon Web Services tramite Okta: da questa pagina del sito del supporto di Okta è possibile ottenere informazioni su come configurare Okta per l'utilizzo con AWS. È possibile configurare Okta per passare i tag di sessione. Per ulteriori informazioni, consulta Okta and AWS Partner to Simplify Access Tramite Session Tags.
Okta	AWS Account Federation: questa sezione del sito Web di Okta descrive come configurare e abilitare IAM Identity Center per. AWS
OneLogin	Nella OneLoginKnowledgebase, **`SAML AWS`** cerca un elenco di articoli che spiegano come configurare la funzionalità di IAM Identity Center tra OneLogin e AWS per scenari a ruolo singolo e multiruolo. È possibile configura re il passaggio dei tag di sessione. OneLogin Per ulteriori informazioni, consulta OneLogin e Tag di sessione: controllo degli accessi alle risorse basato sugli attributi. AWS
Identità Ping	PingFederate AWS Connettore: visualizza i dettagli sul PingFederate AWS Connector, un modello di connessione rapida per configurare facilmente una connessione Single Sign-On (SSO) e di provisioning. Leggi la documenta zione e scarica la versione più recente di PingFederate AWS Connector per le integrazioni con. AWSÈ possibile configurare Ping Identity per passare i tag di sessione. Per ulteriori informazioni, consulta Announcing Ping Identity Support for Attribute-Based Access Control in AWS.
RadiantLogic	Radiant Logic Technology Partners: il RadiantOne Federated Identity Service di Radiant Logic si integra con AWS per fornire un hub di identità per SSO basato su SAML.

Soluzione	Ulteriori informazioni
RSA	Amazon Web Services - RSA Ready Implementation Guide fornisce linee guida per l'integrazione AWS e RSA. Per ulteriori informazioni sulla configurazione SAML, consulta Amazon Web Services - SAML My Page SSO Configuration - RSA Ready Implementation Guide.
Salesforce.com	Come configurare l'SSO da Salesforce a AWS: questo articolo pratico sul sito per sviluppatori Salesforce.com descrive come configurare un provider di identità (IdP) in Salesforce e configurarlo come provider di servizi. AWS
SecureAuth	AWS - SecureAuth SAML SSO: questo articolo sul sito Web descrive come configurare l'integrazione SAML con per un'appliance. SecureAuth AWS SecureAuth
Shibboleth	Come utilizzare Shibboleth per SSO su AWS Management Console: questo articolo del AWS Security Blog fornisce un step-by-step tutorial su come configurare Shibboleth e configurarlo come provider di identità per. AWSÈ possibile configurare Shibboleth per passare i tag di sessione.

Per maggiori dettagli, consulta la pagina IAM Partners sul sito Web. AWS

Configurare le asserzioni SAML per la risposta di autenticazione

Dopo aver verificato l'identità di un utente nella tua organizzazione, il provider di identità esterno (IdP) invia una risposta di autenticazione all'endpoint AWS SAML all'indirizzo. `https://region-code.signin.aws.amazon.com/saml` Per un elenco dei possibili valori di sostituzione di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS. Questa risposta è una richiesta POST che include un token SAML che aderisce al binding POST HTTP per lo standard SAML 2.0 e che contiene i seguenti elementi o attestazioni. È possibile configurare queste affermazioni nell'IDP compatibile con SAML. Per ulteriori informazioni, consulta la documentazione del provider di identità per istruzioni su come inserire queste attestazioni.

Quando l'IdP invia la risposta contenente le attestazioni a AWS, molte delle attestazioni in entrata vengono mappate alle AWS chiavi di contesto. Queste chiavi di contesto possono essere controllate

nelle policy IAM utilizzando l'elemento `Condition`. L'elenco delle mappature disponibili è incluso nella sezione Mappatura degli attributi SAML per considerare attendibili le chiavi contestuali delle AWS politiche.

`Subject` e `NameID`

Di seguito viene riportato un estratto di esempio. Sostituire con i propri valori i valori contrassegnati. Deve essere presente esattamente un elemento `SubjectConfirmation` con un elemento `SubjectConfirmationData` che include sia l'attributo `NotOnOrAfter` che l'attributo `Recipient`. Questi attributi includono un valore che deve corrispondere all' AWS endpoint. `https://`*`region-code`*`.signin.aws.amazon.com/saml` Per un elenco dei possibili valori di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS. Per il AWS valore, puoi anche usare`https://signin.aws.amazon.com/static/saml`, come mostrato nell'esempio seguente.

Gli elementi `NameID` possono avere il valore persistente, transitorio o oppure possono essere costituiti dall'URI formato completo, come fornito dalla soluzione IdP. Un valore permanente indica che il valore in `NameID` è lo stesso per un utente da una sessione all'altra. Se il valore è transitorio, l'utente dispone di un valore `NameID` diverso per ogni sessione. Le interazioni Single Sign-on supportano i seguenti tipi di identificatori:

- `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`
- `urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos`
- `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`

```
<Subject>
  <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">_cbb88bf52c2510eabe00c1642d4643f41430fe25e3</NameID>
  <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <SubjectConfirmationData NotOnOrAfter="2013-11-05T02:06:42.876Z"
 Recipient="https://signin.aws.amazon.com/saml"/>
  </SubjectConfirmation>
```

```
</Subject>
```

> **Important**
>
> Al contrario, la chiave di contesto `saml:aud` proviene dall'attributo recipient SAML perché è l'equivalente SAML del campo del destinatario OIDC, ad esempio, `accounts.google.com:aud`.

Attributo SAML **`PrincipalTag`**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/PrincipalTag:{TagKey}`. Questo elemento consente di passare attributi come tag di sessione nell'asserzione SAML. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.

Per passare gli attributi come tag di sessione, includi l'elemento `AttributeValue` che specifica il valore del tag. Ad esempio, per passare la coppia chiave-valore del tag `Project` = `Marketing` e `CostCenter` = `12345`, utilizza il seguente attributo. Includi un elemento `Attribute` separato per ogni tag.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
  <AttributeValue>Marketing</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:CostCenter">
  <AttributeValue>12345</AttributeValue>
</Attribute>
```

Per impostare i tag sopra elencati come transitivi, includere un altro elemento `Attribute` con l'attributo `Name` impostato a `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`. Questo è un attributo con più valori opzionale che imposta i tag di sessione come transitivi. I tag transitivi persistono quando utilizzi la sessione SAML per assumere un altro ruolo in AWS. Questo è noto come concatenazione del ruolo. Ad esempio, per impostare entrambi i tag `CostCenter` e `Principal` come transitivi, utilizza il seguente attributo per specificare le chiavi.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>Project</AttributeValue>
  <AttributeValue>CostCenter</AttributeValue>
</Attribute>
```

Attributo SAML **`Role`**

Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/Role` Questo elemento contiene uno o più elementi `AttributeValue` che elencano il ruolo e il provider di identità IAM a cui l'utente è mappato dall'IdP. Il ruolo IAM e il provider di identità IAM sono specificati come una coppia di ARN delimitata da virgole nello stesso formato dei `PrincipalArn` parametri `RoleArn` e passati a SAML. AssumeRoleWith Questo elemento deve contenere almeno una coppia ruolo-provider (elemento `AttributeValue`) e può contenere più coppie. Se l'elemento contiene più coppie, all'utente viene chiesto di scegliere il ruolo da assumere quando utilizza WebSSO per accedere alla AWS Management Console.

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/Role`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">
  <AttributeValue>arn:aws:iam::account-number:role/role-name1,arn:aws:iam::account-number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name2,arn:aws:iam::account-number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name3,arn:aws:iam::account-number:saml-provider/provider-name</AttributeValue>
</Attribute>
```

Attributo SAML **`RoleSessionName`**

Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/RoleSessionName` Questo elemento contiene un elemento `AttributeValue` che fornisce un identificatore per le credenziali temporanee emesse quando viene assunto il ruolo. È possibile utilizzare questa opzione per associare le credenziali temporanee all'utente che utilizza l'applicazione. Questo elemento viene utilizzato per visualizzare le informazioni utente in. AWS Management Console Il valore nell'elemento `AttributeValue` deve contenere tra 2 e 64 caratteri, può contenere solo caratteri alfanumerici, caratteri di sottolineatura e i seguenti caratteri: . , + = @ - (trattino). Non può contenere spazi. Il valore è in genere un ID utente (`johndoe`) o un indirizzo e-mail (`johndoe@example.com`). Non deve essere un valore che include uno spazio, ad esempio il nome di visualizzazione di un utente (`John Doe`).

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/RoleSessionName`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/RoleSessionName">
  <AttributeValue>user-id-name</AttributeValue>
</Attribute>
```

Attributo SAML **`SessionDuration`**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SessionDuration"`. Questo elemento contiene un `AttributeValue` elemento che specifica per quanto tempo l'utente può accedere AWS Management Console prima di dover richiedere nuove credenziali temporanee. Il valore è un numero intero che rappresenta il numero di secondi per la sessione. Il valore può variare da 900 secondi (15 minuti) a 43.200 secondi (12 ore). Se questo attributo non è presente, la credenziale dura per un'ora (il valore predefinito del parametro `DurationSeconds` dell'API `AssumeRoleWithSAML`).

Per utilizzare questo attributo, devi configurare il provider SAML in modo che fornisca l'accesso Single Sign-On all'endpoint web di accesso AWS Management Console tramite console all'indirizzo. `https://region-code.signin.aws.amazon.com/saml` Per un elenco dei possibili valori di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS. Facoltativamente, puoi utilizzare il seguente URL: `https://signin.aws.amazon.com/static/saml`. Si noti che questo attributo estende le sessioni solo alla AWS Management Console. Non può estendere la durata di altre credenziali. Tuttavia, se è presente in una chiamata API `AssumeRoleWithSAML`, può essere utilizzato per abbreviare la durata della sessione. La durata predefinita delle credenziali restituite dalla chiamata è di 60 minuti.

Si noti inoltre che, se viene definito anche un attributo `SessionNotOnOrAfter`, il valore inferiore dei due attributi, `SessionDuration` o `SessionNotOnOrAfter`, stabilisce la durata massima della sessione della console.

Quando si abilitano le sessioni della console con una durata estesa, aumenta il rischio di compromissione delle credenziali. Per mitigare questo rischio, è possibile disabilitare immediatamente le sessioni della console attiva per tutti i ruoli, scegliendo Revoca sessioni nella

pagina Riepilogo ruolo della console IAM. Per ulteriori informazioni, consulta Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

> **Important**
>
> Il valore dell'attributo `Name` nel tag `Attribute` è sensibile alla distinzione tra maiuscolo/minuscolo. Il valore deve essere impostato esattamente su `https://aws.amazon.com/SAML/Attributes/SessionDuration`.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SessionDuration">
  <AttributeValue>1800</AttributeValue>
</Attribute>
```

Attributo SAML **`SourceIdentity`**

(Facoltativo) Puoi utilizzare un elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SourceIdentity`. Questo elemento contiene un elemento `AttributeValue` che fornisce un identificatore per la persona o l'applicazione che utilizza un ruolo IAM. Il valore dell'identità di origine persiste quando utilizzi la sessione SAML per assumere un altro ruolo, noto come concatenamento dei ruoli. AWS Il valore per l'identità di origine è presente nella richiesta per ogni operazione eseguita durante la sessione del ruolo. Il valore impostato non può essere modificato durante la sessione del ruolo. Gli amministratori possono quindi utilizzare AWS CloudTrail i log per monitorare e controllare le informazioni sull'identità di origine per determinare chi ha eseguito azioni con ruoli condivisi.

Il valore nell'elemento `AttributeValue` deve contenere tra 2 e 64 caratteri, può contenere solo caratteri alfanumerici, caratteri di sottolineatura e i seguenti caratteri: . , + = @ - (trattino). Non può contenere spazi. Il valore è in genere un attributo associato all'utente, ad esempio un ID utente (`johndoe`) o un indirizzo e-mail (`johndoe@example.com`). Non deve essere un valore che include uno spazio, ad esempio il nome di visualizzazione di un utente (`John Doe`). Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

> **Important**
>
> Se l'asserzione SAML è configurata per utilizzare l'attributo `SourceIdentity`, allora la policy di attendibilità del ruolo deve includere anche l'operazione `sts:SetSourceIdentity`

altrimenti l'operazione di assunzione del ruolo avrà esito negativo. Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Per inviare un attributo dell'identità di origine, includi l'elemento `AttributeValue` che specifica il valore dell'identità di origine. Ad esempio, per inviare `DiegoRamirez` dell'identità di origine, utilizza il seguente attributo.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SourceIdentity">
  <AttributeValue>DiegoRamirez</AttributeValue>
```

Mappatura degli attributi SAML per considerare attendibili le chiavi contestuali delle AWS politiche

Le tabelle in questa sezione elencano gli attributi SAML comunemente usati e il modo in cui sono mappati alle chiavi di contesto delle condizioni delle policy in AWS. Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi con i valori che sono inclusi nelle asserzioni incluse in una richiesta di accesso SAML.

Important

Queste chiavi sono disponibili solo nelle policy di affidabilità IAM (policy che determinano chi può assumere un ruolo) e non sono applicabili alle policy di autorizzazione.

Nella tabella degli attributi eduPerson e eduOrg, i valori vengono digitati come stringhe o come elenchi di stringhe. Per i valori di stringa, puoi testare questi valori nelle policy di attendibilità IAM utilizzando le condizioni `StringEquals` o `StringLike`. Per i valori che contengono un elenco di stringhe, è possibile utilizzare gli `ForAnyValue`operatori del set di policy`ForAllValues` e per un test dei valori delle policy di attendibilità.

Note

Dovresti includere solo un claim per chiave di AWS contesto. Se ne include più di una, verrà mappata una sola attestazione.

Attributi eduPerson ed eduOrg

Attributo eduPerson o eduOrg (chiave **`Name`**)	Si associa a questa chiave di AWS contesto (**`FriendlyName`** chiave)	Type
`urn:oid:1.3.6.1.4.1.5923.1.1.1.1`	`eduPersonAffiliation`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.2`	`eduPersonNickname`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.3`	`eduPersonOrgDN`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.4`	`eduPersonOrgUnitDN`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.5`	`eduPersonPrimaryAffiliation`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.6`	`eduPersonPrincipalName`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.7`	`eduPersonEntitlement`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.8`	`eduPersonPrimaryOrgUnitDN`	Stringa
`urn:oid:1.3.6.1.4.1.5923.1.1.1.9`	`eduPersonScopedAffiliation`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.10`	`eduPersonTargetedID`	Elenco di stringhe
`urn:oid:1.3.6.1.4.1.5923.1.1.1.11`	`eduPersonAssurance`	Elenco di stringhe

Attributo eduPerson o eduOrg (chiave **Name**)	Si associa a questa chiave di AWS contesto (**FriendlyName** chiave)	Type
urn:oid:1.3.6.1.4.1.5923.1.2.1.2	eduOrgHomePageURI	Elenco di stringhe
urn:oid:1.3.6.1.4.1.5923.1.2.1.3	eduOrgIdentityAuthNPolicyURI	Elenco di stringhe
urn:oid:1.3.6.1.4.1.5923.1.2.1.4	eduOrgLegalName	Elenco di stringhe
urn:oid:1.3.6.1.4.1.5923.1.2.1.5	eduOrgSuperiorURI	Elenco di stringhe
urn:oid:1.3.6.1.4.1.5923.1.2.1.6	eduOrgWhitePagesURI	Elenco di stringhe
urn:oid:2.5.4.3	cn	Elenco di stringhe

Attributi di Active Directory

Attributo AD	Si associa a questa chiave di AWS contesto	Type
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	name	Stringa
http://schemas.xmlsoap.org/claims/CommonName	commonName	Stringa
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	givenName	Stringa
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	surname	Stringa

Attributo AD	Si associa a questa chiave di AWS contesto	Type
`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`	`mail`	Stringa
`http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid`	`uid`	Stringa

Attributi X.500

Attributo X.500	Si associa a questa chiave di AWS contesto	Type
`2.5.4.3`	`commonName`	Stringa
`2.5.4.4`	`surname`	Stringa
`2.4.5.42`	`givenName`	Stringa
`2.5.4.45`	`x500UniqueIdentifier`	Stringa
`0.9.2342.19200300100.1.1`	`uid`	Stringa
`0.9.2342.19200300100.1.3`	`mail`	Stringa
`0.9.2342.19200300.100.1.45`	`organizationStatus`	Stringa

Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console

Puoi utilizzare un ruolo per configurare il tuo provider di identità (IdP) conforme a SAML 2.0 e consentire AWS agli utenti federati di accedere a. AWS Management Console Il ruolo concede all'utente le autorizzazioni per eseguire attività nella console. Se invece desideri fornire agli utenti federati SAML altri metodi per accedere ad AWS, consulta uno dei seguenti argomenti:

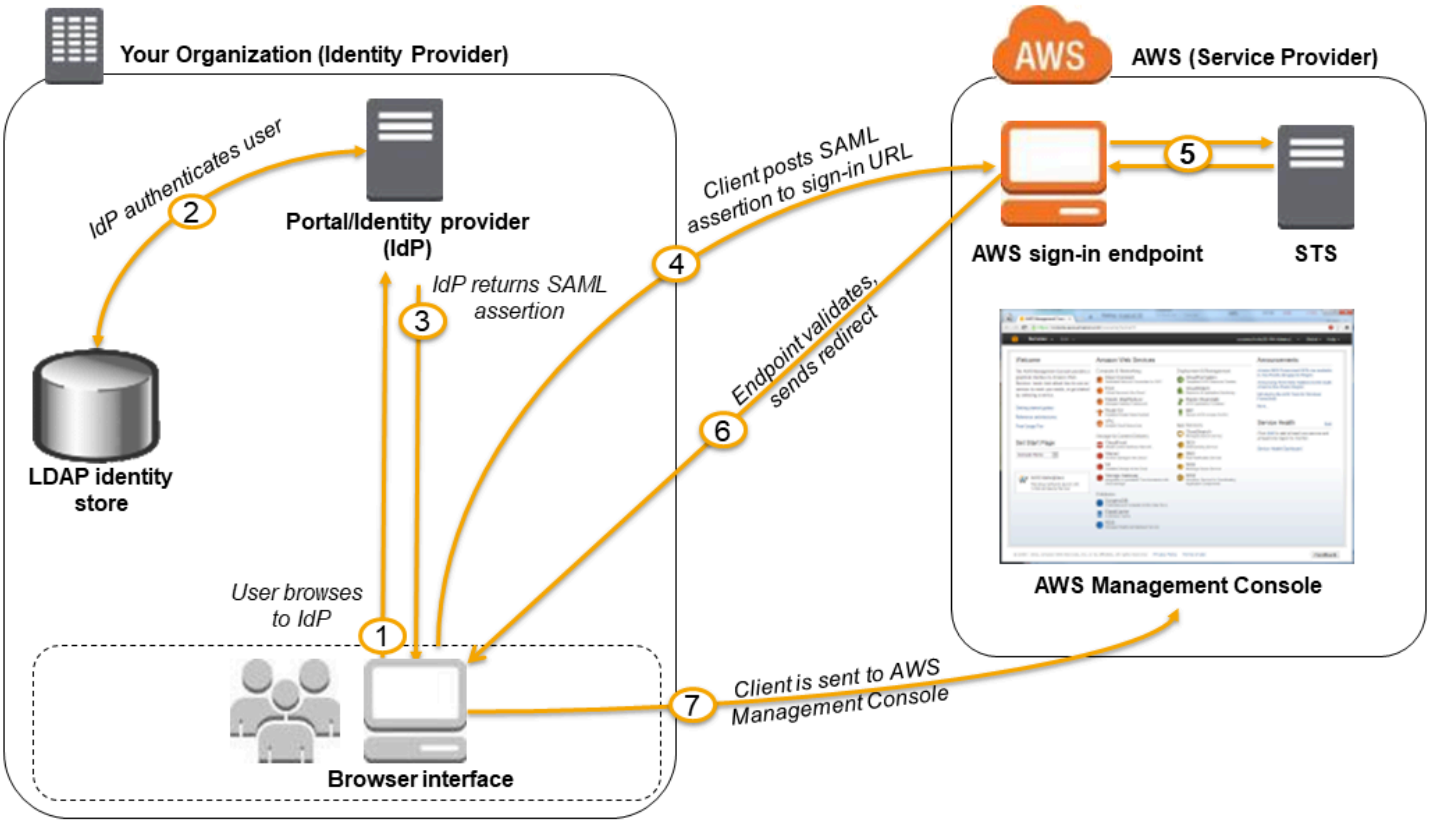
- AWS CLI: Passaggio a un ruolo IAM (AWS CLI)

- Strumenti per Windows: PowerShell Passaggio a un ruolo IAM (Tools for Windows PowerShell)
- AWS API: Passaggio a un ruolo IAM (AWS API)

Panoramica

Il seguente diagramma mostra il flusso per il Single Sign-On abilitato per SAML.

> **Note**
>
> Questo uso specifico di SAML differisce da quello più generale illustrato in precedenza Federazione SAML 2.0 perché questo flusso di lavoro lo apre per AWS Management Console conto dell'utente. In questo caso occorre utilizzare l'endpoint di accesso ad AWS anziché richiamare direttamente l'API `AssumeRoleWithSAML`. L'endpoint richiama l'API per l'utente e restituisce un'URL che reindirizza automaticamente il browser dell'utente alla AWS Management Console.



Il diagramma illustra i passaggi seguenti:

1. L'utente accede al portale dell'organizzazione e seleziona l'opzione che consente di accedere alla AWS Management Console. Nella tua organizzazione, il portale è in genere una funzione del tuo IdP che gestisce lo scambio di fiducia tra l'organizzazione e. AWS Ad esempio, in Active Directory Federation Services, l'URL del portale è: `https://ADFSServiceName/adfs/ls/IdpInitiatedSignOn.aspx`
2. Il portale verifica l'identità dell'utente nell'organizzazione.
3. Il portale genera una risposta di autenticazione SAML che include asserzioni che identificano l'utente e includono gli attributi dell'utente. È anche possibile configurare il provider di identità per includere un attributo di asserzione SAML chiamato `SessionDuration` che specifica la durata della validità della sessione della console. È anche possibile configurare il provider di identità per passare gli attributi come tag di sessione. Il portale invia questa risposta al browser del client.
4. Il browser del client viene reindirizzato all'endpoint AWS Single Sign-On e pubblica l'asserzione SAML.
5. L'endpoint richiede le credenziali di sicurezza provvisorie per conto dell'utente e crea una URL di accesso alla console che utilizza tali credenziali.
6. AWS invia l'URL di accesso al client come reindirizzamento.
7. Il browser del client è reindirizzato alla AWS Management Console. Se la risposta di autenticazione SAML include attributi mappati a più ruoli IAM, all'utente viene chiesto di selezionare il ruolo per l'accesso alla console.

Dal punto di vista dell'utente, il processo avviene in modo trasparente: l'utente inizia dal portale interno dell'organizzazione e finisce al AWS Management Console, senza mai dover fornire alcuna credenziale. AWS

Consulta le seguenti sezioni per una panoramica della configurazione di questo comportamento insieme ai collegamenti alla procedura dettagliata.

Configura la tua rete come provider SAML per AWS

All'interno della rete aziendale, è possibile configurare l'archivio identità (ad esempio Windows Active Directory) per l'utilizzo con un IdP basato su SAML come, ad esempio, Windows Active Directory Federation Services e Shibboleth. Utilizzando il provider di identità, si genera un documento di metadati che descrive l'organizzazione come un IdP e include le chiavi di autenticazione. Inoltre, configuri il portale della tua organizzazione per indirizzare le richieste degli utenti AWS Management Console all'endpoint AWS SAML per l'autenticazione utilizzando le asserzioni SAML. Il modo in cui è possibile configurare il provider di identità per la produzione del file metadata.xml dipende dal

provider di identità. Per ulteriori informazioni, consulta la documentazione del provider di identità, oppure consulta Integra fornitori di soluzioni SAML di terze parti con AWS per i collegamenti alla documentazione Web per molti dei fornitori di SAML supportati.

Creazione di un provider SAML in IAM

Successivamente, accedi AWS Management Console e vai alla console IAM. Qui si crea un nuovo provider SAML, ovvero un'entità in IAM che contiene informazioni sul provider di identità dell'organizzazione. Come parte di questo processo, è possibile caricare il documento di metadati prodotto dal software IdP nella propria organizzazione nella sezione precedente. Per informazioni dettagliate, vedi Crea un provider di identità SAML in IAM.

Configura le autorizzazioni AWS per i tuoi utenti federati

La fase successiva consiste nel creare un ruolo IAM che stabilisca una relazione di attendibilità tra IAM e il provider di identità dell'organizzazione. Questo ruolo deve identificare il tuo provider di identità come un principale (entità attendibile) ai fini della federazione. Il ruolo definisce anche le operazioni consentite agli utenti autenticati dall'IdP dell'organizzazione. AWSÈ possibile utilizzare la console IAM per creare questo ruolo. Quando si crea la policy di attendibilità che indica chi può assumere il ruolo, specifica il provider SAML creato in precedenza in IAM. È inoltre possibile specificare uno o più attributi SAML a cui un utente deve corrispondere per poter assumere quel ruolo. Ad esempio, è possibile specificare che solo gli utenti il cui valore SAML `eduPersonOrgDN` è `ExampleOrg` sono autorizzati ad accedere. La procedura guidata relativa al ruolo aggiunge automaticamente una condizione per testare l'attributo `saml:aud` per assicurarsi che il ruolo venga assunto solo per l'accesso alla AWS Management Console. La policy di affidabilità potrebbe apparire come segue:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"Federated": "arn:aws:iam::account-id:saml-provider/
ExampleOrgSSOProvider"},
    "Action": "sts:AssumeRoleWithSAML",
    "Condition": {"StringEquals": {
      "saml:edupersonorgdn": "ExampleOrg",
      "saml:aud": "https://signin.aws.amazon.com/saml"
    }}
  }]
}
```

> **Note**
>
> Gli IDP SAML utilizzati in una policy di attendibilità dei ruoli devono appartenere allo stesso account in cui si trova il ruolo.

Puoi includere endpoint regionali per l'attributo `saml:aud` in `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml`. Per un elenco dei possibili valori di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS.

Per la policy di autorizzazione nel ruolo, è necessario specificare le autorizzazioni come per qualsiasi utente, gruppo o ruolo. Ad esempio, se gli utenti dell'organizzazione sono autorizzati ad amministrare le istanze Amazon EC2, consenti esplicitamente le operazioni Amazon EC2 nella policy di autorizzazione. A tale scopo, puoi assegnare una policy gestita, ad esempio la policy di accesso completo di Amazon EC2.

Per ulteriori informazioni sulla creazione di un ruolo per un provider di identità SAML, consulta Creare un ruolo per la federazione SAML 2.0 (console).

Fine della configurazione e creazione di asserzioni SAML

Informa il tuo IdP SAML AWS che è il tuo fornitore di servizi installando `saml-metadata.xml` il file disponibile `https://region-code.signin.aws.amazon.com/static/saml-metadata.xml` in o. `https://signin.aws.amazon.com/static/saml-metadata.xml` Per un elenco dei possibili valori di *`region-code`*, consulta la colonna Region (Regione) in Endpoint di accesso AWS.

Il modo in cui installare tale file dipende dal provider di identità. Alcuni provider danno la possibilità di digitare l'URL, dopodiché il provider di identità ottiene e installa il file per conto dell'utente. Altri richiedono di scaricare il file dall'URL e quindi fornirlo come file locale. Per ulteriori informazioni, consulta la documentazione del provider di identità, oppure consulta Integra fornitori di soluzioni SAML di terze parti con AWS per i collegamenti alla documentazione Web per molti dei fornitori di SAML supportati.

È anche possibile configurare le informazioni che si desidera che il provider di identità passi come attributi SAML per AWS come parte della risposta di autenticazione. La maggior parte di queste informazioni viene visualizzata AWS come chiavi di contesto delle condizioni che puoi valutare nelle tue politiche. Queste chiavi di condizione garantiscono che solo agli utenti autorizzati nei giusti contesti vengono concesse le autorizzazioni per accedere alle risorse AWS . È possibile specificare le finestre di tempo che limitano l'utilizzo della console. È inoltre possibile specificare il tempo

massimo (fino a 12 ore) durante il quale gli utenti possono accedere alla console prima di dover aggiornare le proprie credenziali. Per informazioni dettagliate, vedi Configurare le asserzioni SAML per la risposta di autenticazione.

Credenziali di sicurezza temporanee in IAM

Puoi utilizzare AWS Security Token Service (AWS STS) per creare e fornire agli utenti attendibili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue AWS risorse. Le credenziali di sicurezza temporanee funzionano quasi esattamente come le credenziali delle chiavi di accesso a lungo termine, con le seguenti differenze:

- Le credenziali di sicurezza provvisorie sono a breve termine, come implica il nome. Possono essere configurate per durare ovunque per pochi minuti o diverse ore. Una volta scadute, le credenziali AWS non le riconosce più né consente alcun tipo di accesso alle richieste API effettuate con esse.
- Le credenziali di sicurezza temporanee non sono archiviate con l'utente, ma vengono generate dinamicamente e fornite all'utente quando richiesto. Quando (o anche prima) le credenziali di sicurezza temporanee scadono, l'utente può richiedere nuove credenziali, purché l'utente che le richiede abbia ancora le autorizzazioni per farlo.

Di conseguenza, le credenziali temporanee presentano i seguenti vantaggi rispetto alle credenziali a lungo termine:

- Non è necessario distribuire o incorporare credenziali di AWS sicurezza a lungo termine in un'applicazione.
- È possibile fornire l'accesso alle AWS risorse agli utenti senza dover definire un' AWS identità per loro. Le credenziali temporanee sono la base per la federazione dei ruoli e delle identità.
- Le credenziali di sicurezza temporanee hanno una durata limitata, perciò non è necessario aggiornarle o revocarle in modo esplicito quando non sono più necessarie. Dopo che le credenziali di sicurezza temporanee scadono, non possono essere riutilizzate. È possibile specificare quando scadono le credenziali, fino a un limite massimo.

AWS STS e regioni AWS

Le credenziali di sicurezza temporanee sono generate da AWS STS. Per impostazione predefinita, AWS STS è un servizio globale con un unico endpoint su`https://sts.amazonaws.com`. Tuttavia,

puoi anche scegliere di effettuare chiamate AWS STS API verso endpoint in qualsiasi altra regione supportata. Ciò può ridurre la latenza (server lag) effettuando le richieste a server in una regione geograficamente più vicina a te. Indipendentemente dalla regione dalla quale provengono, le credenziali funzionano a livello globale. Per ulteriori informazioni, consulta Gestire AWS STS in un Regione AWS.

Scenari comuni per le credenziali temporanee

Le credenziali temporanee sono utili in scenari che interessano la federazione delle identità, la delega, l'accesso tra account e i ruoli IAM.

Federazione delle identità

Puoi gestire le tue identità utente in un sistema esterno esterno AWS e concedere agli utenti che accedono da tali sistemi l'accesso per eseguire AWS attività e accedere alle tue AWS risorse. IAM supporta due tipi di federazione delle identità. In entrambi i casi, le identità vengono archiviate all'esterno di. AWS La differenza è dove risiede il sistema esterno: nel data center o una parte terza sul Web. Per ulteriori informazioni sui provider di identità esterni, consultare Provider di identità e federazione.

- Federazione SAML: puoi autenticare gli utenti nella rete dell'organizzazione e quindi fornire loro l'accesso AWS senza creare nuove AWS identità per loro e richiedere loro di accedere con credenziali di accesso diverse. Questo è noto come approccio Single Sign-On all'accesso temporaneo. AWS STS supporta standard aperti come Security Assertion Markup Language (SAML) 2.0, con cui è possibile utilizzare Microsoft AD FS per sfruttare Microsoft Active Directory. È inoltre possibile utilizzare SAML 2.0 per gestire la soluzione per la federazione delle identità dell'utente. Per ulteriori informazioni, consulta Federazione SAML 2.0.
  - Broker federativo personalizzato: puoi utilizzare il sistema di autenticazione della tua organizzazione per concedere l'accesso alle risorse. AWS Per uno scenario di esempio, consultare Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.
  - Federazione tramite SAML 2.0: puoi utilizzare SAML e il sistema di autenticazione dell'organizzazione per concedere l'accesso alle risorse AWS . Per ulteriori informazioni e uno scenario di esempio, consultare Federazione SAML 2.0.
- Federazione OpenID Connect (OIDC): puoi consentire agli utenti di accedere utilizzando un noto provider di identità di terze parti come Login with Amazon, Facebook, Google o qualsiasi provider compatibile con OIDC 2.0 per la tua applicazione mobile o web, non devi creare un codice di

accesso personalizzato o gestire le tue identità utente. L'utilizzo della federazione OIDC ti aiuta a mantenere la tua Account AWS sicurezza, perché non devi distribuire credenziali di sicurezza a lungo termine, come le chiavi di accesso utente IAM, con la tua applicazione. Per ulteriori informazioni, consulta Federazione OIDC.

AWS STS La federazione OIDC supporta Login with Amazon, Facebook, Google e qualsiasi provider di identità compatibile con OpenID Connect (OIDC).

> **Note**
>
> Per le applicazioni mobili, consigliamo di utilizzare Amazon Cognito. Puoi utilizzare questo servizio con gli AWS SDK per lo sviluppo mobile per creare identità uniche per gli utenti e autenticarle per un accesso sicuro alle tue risorse. AWS Amazon Cognito supporta gli stessi provider di identità e supporta anche l'accesso non autenticato (guest) e consente di migrare i dati degli utenti quando un utente accede. AWS STS Amazon Cognito fornisce inoltre operazioni API per la sincronizzazione dei dati utente in modo che vengano conservati quando gli utenti passano da un dispositivo all'altro. Per ulteriori informazioni, consulta Autenticazione con Amplify nella documentazione di Amplify.

Ruoli per l'accesso tra account

Molte organizzazioni mantengono più di un Account AWS. Utilizzando ruoli e l'accesso tra account, è possibile definire le identità degli utenti in un account e utilizzare tali identità per accedere alle risorse AWS in altri account che appartengono all'organizzazione. Questo approccio è noto come delega all'accesso temporaneo. Per ulteriori informazioni sulla creazione di ruoli tra account, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un utente IAM. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Ruoli per Amazon EC2

Se esegui applicazioni sulle istanze Amazon EC2 e tali applicazioni devono accedere alle risorse AWS , puoi fornire le credenziali di sicurezza temporanee alle istanze al momento dell'avvio. Queste credenziali di sicurezza temporanee sono disponibili a tutte le applicazioni che vengono eseguite sull'istanza, perciò non è necessario archiviare nessuna delle credenziali a lungo termine sull'istanza. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Altri servizi AWS

È possibile utilizzare credenziali di sicurezza temporanee per accedere alla maggior parte dei AWS servizi. Per un elenco dei servizi che accettano le credenziali di sicurezza temporanee, consultare AWS servizi che funzionano con IAM.

Richiesta di credenziali di sicurezza temporanee

Per richiedere credenziali di sicurezza temporanee, puoi utilizzare le operazioni AWS Security Token Service (AWS STS) nell' AWS API. Queste includono operazioni per creare e fornire a utenti affidabili credenziali di sicurezza temporanee in grado di controllare l'accesso alle AWS risorse. Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. Per informazioni sui diversi metodi che si possono utilizzare per richiedere credenziali di sicurezza temporanee assumendo un ruolo, consultare Utilizzo di ruoli IAM.

Per chiamare le operazioni API, è possibile utilizzare uno degli SDK AWS. Gli SDK sono disponibili per un'ampia gamma di linguaggi di programmazione e ambienti, tra cui Java, .NET, Python, Ruby, Android e iOS. I kit SDK si occupano di attività quali la firma crittografica delle richieste, la ripetizione delle richieste se necessario e la gestione delle risposte agli errori. Puoi anche utilizzare l'API AWS STS Query, descritta nel riferimento all'AWS Security Token Service API. Infine, due strumenti da riga di comando supportano i AWS STS comandi: the AWS Command Line Interface, e the AWS Tools for Windows PowerShell.

Le operazioni AWS STS API creano una nuova sessione con credenziali di sicurezza temporanee che includono una coppia di chiavi di accesso e un token di sessione. La coppia di chiavi di accesso è composta da un ID chiave di accesso e da una chiave segreta. Gli utenti (o un'applicazione che l'utente esegue) possono utilizzare queste credenziali per accedere alle risorse. È possibile creare una sessione di ruolo e passare policy di sessione e tag di sessione in modo programmatico utilizzando AWS STS le operazioni API. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Per ulteriori informazioni sulle policy di sessione, consulta Policy di sessione. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.

> **Note**
>
> La dimensione del token di sessione restituito dalle operazioni AWS STS API non è fissa. È consigliabile di non effettuare alcuna supposizione sulle dimensioni massime. La dimensione tipica dei token è inferiore a 4.096 byte, ma essa può variare.

Utilizzo AWS STS con AWS le regioni

Puoi inviare chiamate AWS STS API a un endpoint globale o a uno degli endpoint regionali. Se si seleziona un endpoint vicino, è possibile ridurre la latenza e migliorare le prestazioni delle chiamate API. Se non è più possibile comunicare con l'endpoint originale è anche possibile scegliere di indirizzare le chiamate verso un endpoint regionale alternativo. Se utilizzi uno dei vari AWS SDK, utilizza quel metodo SDK per specificare una regione prima di effettuare la chiamata API. Se costruiscono manualmente richieste API HTTP, è necessario indirizzare la richiesta all'endpoint corretto. Per ulteriori informazioni, consulta la sezione AWS STS relativa alle regioni e agli endpoint e la sezione Gestire AWS STS in un Regione AWS.

Di seguito sono elencate le operazioni API che puoi utilizzare per acquisire credenziali temporanee da utilizzare nell' AWS ambiente e nelle applicazioni.

AssumeRole: delega tra account e federazione tramite un gestore identità personalizzato

L'operazione `AssumeRole` API è utile per consentire agli utenti IAM esistenti di accedere a AWS risorse a cui non hanno già accesso. Ad esempio, l'utente potrebbe aver bisogno di accedere alle risorse in un altro Account AWS. Inoltre, è utile come strumento per ottenere temporaneamente l'accesso privilegiato, ad esempio per fornire l'autenticazione a più fattori (MFA). È necessario chiamare questa API utilizzando le credenziali attive. Per sapere chi può chiamare questa operazione, vedere Confronto delle operazioni AWS STS API. Per ulteriori informazioni, consulta Creazione di un ruolo per delegare le autorizzazioni a un utente IAM e Configurazione dell'accesso alle API protetto da MFA.

Questa chiamata deve essere effettuata utilizzando credenziali AWS di sicurezza valide. Quando si effettua questa chiamata, vengono fornite le seguenti informazioni:

- L'Amazon Resource Name (ARN) del ruolo che deve assumere l'app.
- (Facoltativo) Durata, che specifica la durata delle credenziali di sicurezza temporanee. Utilizzare il parametro `DurationSeconds` per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo. Se non si passa questo parametro, le credenziali temporanee scadono in un'ora. Il parametro `DurationSeconds` da questa API è separato dal parametro HTTP `SessionDuration` che viene utilizzato per specificare la durata di una sessione di console. Utilizzare il parametro HTTP `SessionDuration` nella richiesta all'endpoint

di federazione per un token di accesso alla console. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

- Nome della sessione del ruolo Utilizza questo valore stringa per identificare la sessione quando un ruolo viene utilizzato da principali diversi. Per motivi di sicurezza, gli amministratori possono visualizzare questo campo nei log AWS CloudTrail per sapere chi ha eseguito un'operazione in AWS. L'amministratore potrebbe richiedere di specificare il nome utente IAM come nome di sessione quando assumi il ruolo. Per ulteriori informazioni, consulta `sts:RoleSessionName`.
- (Facoltativo) Identità di origine. È possibile richiedere agli utenti di specificare un'identità di origine quando assumono un ruolo. Dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nella richiesta di tutte le operazioni intraprese durante la sessione del ruolo. Il valore dell'identità di origine persiste tra le varie sessioni di ruolo concatenati. È possibile utilizzare le informazioni sull'identità di origine nei AWS CloudTrail log per determinare chi ha intrapreso azioni con un ruolo. Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.
- (Facoltativo) Policy di gestione inline o gestite. Queste policy limitano le autorizzazioni derivate dalla policy basata sull'identità del ruolo che sono assegnate alla sessione del ruolo. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.
- (Facoltativo) Tag di sessione. È possibile assumere un ruolo e quindi utilizzare le credenziali temporanee per effettuare una richiesta. In questo caso, i tag di entità della sessione includono i tag del ruolo e i tag di sessione passati. Se si effettua questa chiamata utilizzando credenziali temporanee, la nuova sessione eredita anche i tag di sessione transitivi della sessione chiamante. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.
- (Facoltativo) Informazioni su MFA. Se configurato per utilizzare la multi-factor authentication (MFA), includere l'identificatore per un dispositivo MFA e il singolo codice fornito da quel dispositivo.
- (Facoltativo) Valore `ExternalId` che può essere utilizzato quando si delega l'accesso all'account a terze parti. Questo valore contribuisce ad assicurare che solo la terza parte specificata possa accedere al ruolo. Per ulteriori informazioni, consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.

L'esempio seguente mostra una richiesta di esempio e la risposta utilizzando `AssumeRole`. Questa richiesta di esempio assume il ruolo demo per la durata specificata, inclusi policy di sessione, tag di sessione, ID esterno e identità di origine. La sessione risultante è denominata `John-session`.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=AssumeRole
&RoleSessionName=John-session
&RoleArn=arn:aws::iam::123456789012:role/demo
&Policy=%7B%22Version%22%3A%222012-10-17%22%2C%22Statement%22%3A%5B%7B%22Sid%22%3A
%20%22Stmt1%22%2C%22Effect%22%3A%20%22Allow%22%2C%22Action%22%3A%20%22s3%3A*%22%2C
%22Resource%22%3A%20%22*%22%7D%5D%7D
&DurationSeconds=1800
&Tags.member.1.Key=Project
&Tags.member.1.Value=Pegasus
&Tags.member.2.Key=Cost-Center
&Tags.member.2.Value=12345
&ExternalId=123ABC
&SourceIdentity=DevUser123
&AUTHPARAMS
```

Il valore della policy illustrato nell'esempio precedente è la versione di codifica URL della policy seguente:

```
{"Version":"2012-10-17","Statement":
[{"Sid":"Stmt1","Effect":"Allow","Action":"s3:*","Resource":"*"}]}
```

Il parametro `AUTHPARAMS` nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che è necessario includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, consulta Firmare AWS le richieste utilizzando la versione 4 della Riferimenti generali di Amazon Web Servicespagina per scoprire come firmare una richiesta.

Oltre alle credenziali di sicurezza temporanee, la risposta include l'Amazon Resource Name (ARN) per l'utente federato e il periodo di scadenza delle credenziali.

Example Example response

```
<AssumeRoleResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
```

```
<AssumeRoleResult>
<SourceIdentity>DevUser123</SourceIdentity>
<Credentials>
  <SessionToken>
   AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
   LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
   QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
   9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
   +scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==
  </SessionToken>
  <SecretAccessKey>
   wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2019-07-15T23:28:33.359Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
<AssumedRoleUser>
  <Arn>arn:aws:sts::123456789012:assumed-role/demo/John</Arn>
  <AssumedRoleId>ARO123EXAMPLE123:John</AssumedRoleId>
</AssumedRoleUser>
<PackedPolicySize>8</PackedPolicySize>
</AssumeRoleResult>
<ResponseMetadata>
<RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
</ResponseMetadata>
</AssumeRoleResponse>
```

> **Note**
>
> Una AWS conversione comprime le politiche di sessione e i tag di sessione passati in un formato binario compresso con un limite separato. La richiesta può non essere eseguita correttamente a causa di questo limite anche se il testo in chiaro soddisfa gli altri requisiti. L'elemento della risposta `PackedPolicySize` indica in percentuale la consistenza di policy e tag della richiesta rispetto al limite di dimensione superiore.

AssumeRoleWithWebIdentity: federazione tramite un provider di identità basato sul Web

L'operazione API `AssumeRoleWithWebIdentity` restituisce un set di credenziali di sicurezza temporanee per gli utenti federati che sono autenticati attraverso un provider di identità pubblico.

Esempi di provider di identità pubblici comprendono Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Questa operazione è utile per creare applicazioni mobili o applicazioni Web basate su client che richiedono l'accesso a. AWS L'utilizzo di questa operazione significa che gli utenti non hanno bisogno della propria identità AWS o di quella di IAM. Per ulteriori informazioni, consulta Federazione OIDC.

Invece di chiamare direttamente`AssumeRoleWithWebIdentity`, ti consigliamo di utilizzare Amazon Cognito e il provider di credenziali Amazon Cognito con gli SDK per AWS lo sviluppo mobile. Per ulteriori informazioni, consulta Autenticazione con Amplify nella documentazione di Amplify.

Se non utilizzi Amazon Cognito, richiama l'operazione `AssumeRoleWithWebIdentity` di AWS STS. Si tratta di una chiamata non firmata, il che significa che l'app non ha bisogno di accedere a nessuna credenziale di sicurezza di AWS per effettuare la chiamata. Quando si effettua questa chiamata, vengono fornite le seguenti informazioni:

- L'Amazon Resource Name (ARN) del ruolo che deve assumere l'app. Se l'app supporta diversi modi per gli utenti per effettuare l'accesso, è necessario definire più ruoli, uno per ogni provider di identità. La chiamata a `AssumeRoleWithWebIdentity` deve includere l'ARN del ruolo specifico per il provider attraverso il quale l'utente ha effettuato l'accesso.
- Il token che l'applicazione ottiene dal provider di identità dopo che l'app autentica l'utente.
- È possibile configurare il proprio IdP per passare attributi nel proprio token come tag di sessione.
- (Facoltativo) Durata, che specifica la durata delle credenziali di sicurezza temporanee. Utilizzare il parametro `DurationSeconds` per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo. Se non si passa questo parametro, le credenziali temporanee scadono in un'ora. Il parametro `DurationSeconds` da questa API è separato dal parametro HTTP `SessionDuration` che viene utilizzato per specificare la durata di una sessione di console. Utilizzare il parametro HTTP `SessionDuration` nella richiesta all'endpoint di federazione per un token di accesso alla console. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.
- Nome della sessione del ruolo Utilizza questo valore stringa per identificare la sessione quando un ruolo viene utilizzato da principali diversi. Per motivi di sicurezza, gli amministratori possono visualizzare questo campo nei log AWS CloudTrail per sapere chi ha eseguito un'operazione in AWS. L'amministratore potrebbe richiedere di fornire un valore specifico per il nome della sessione quando si assume il ruolo. Per ulteriori informazioni, consulta `sts:RoleSessionName`.

- (Facoltativo) Identità di origine. Puoi richiedere agli utenti federati di specificare un'identità di origine quando assumono un ruolo. Dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nella richiesta di tutte le operazioni intraprese durante la sessione del ruolo. Il valore dell'identità di origine persiste tra le varie sessioni di ruolo concatenati. Puoi utilizzare le informazioni sull'identità di origine nei AWS CloudTrail log per determinare chi ha intrapreso azioni con un ruolo. Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.
- (Facoltativo) Policy di gestione inline o gestite. Queste policy limitano le autorizzazioni derivate dalla policy basata sull'identità del ruolo che sono assegnate alla sessione del ruolo. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

> **Note**
>
> Una chiamata a `AssumeRoleWithWebIdentity` non è firmata (crittografata). Pertanto, è opportuno includere le policy di sessione facoltative solo se la richiesta viene trasmessa attraverso un intermediario affidabile. In questo caso, qualcuno potrebbe modificare la policy per rimuovere le limitazioni.

Quando si chiama`AssumeRoleWithWebIdentity`, AWS verifica l'autenticità del token. Ad esempio, a seconda del provider, AWS potrebbe effettuare una chiamata al provider e includere il token passato dall'app. Supponendo che il provider di identità convalidi il token, ti AWS restituisce le seguenti informazioni:

- Un set di credenziali di sicurezza temporanee. Consistono in un ID chiave di accesso, in una Secret Access Key e in un token di sessione.
- l'ID del ruolo e l'ARN del ruolo assunto.
- Un valore `SubjectFromWebIdentityToken` che contiene l'ID utente univoco.

Se disponi delle credenziali di sicurezza temporanee, puoi utilizzarle per effettuare AWS chiamate API. Si tratta dello stesso processo utilizzato per effettuare una chiamata AWS API con credenziali

di sicurezza a lungo termine. La differenza è che è necessario includere il token della sessione, che consente ad AWS di verificare che le credenziali di sicurezza provvisorie siano valide.

L'app deve memorizzare le credenziali. Come specificato, come impostazione predefinita le credenziali scadono dopo un'ora. Se non utilizzi l'CredentialsProvideroperazione AmazonSTS nell' AWS SDK, spetta a te e alla tua app effettuare nuovamente la chiamata. `AssumeRoleWithWebIdentity` Chiamare questa operazione per ottenere un nuovo set di credenziali di sicurezza provvisorie prima che i vecchi scadere.

AssumeRoleWithSAML: federazione tramite un provider di identità aziendale compatibile con SAML 2.0

L'operazione API `AssumeRoleWithSAML` restituisce un set di credenziali di sicurezza temporanee per gli utenti federati che sono autenticati dal sistema di identità esistente dell'organizzazione. Gli utenti devono utilizzare inoltre SAML 2.0 (Security Assertion Markup Language) per inoltrare le informazioni di autenticazione e autorizzazione ad AWS. Questa operazione di API è utile in organizzazioni che hanno integrato i propri sistemi di identità (ad esempio Windows Active Directory o OpenLDAP) con software che è in grado di produrre asserzioni SAML. Tale integrazione fornisce informazioni sulle autorizzazioni e identità dell'utente (ad esempio Active Directory Federation Services o Shibboleth). Per ulteriori informazioni, consulta Federazione SAML 2.0.

> **Note**
>
> Una chiamata a `AssumeRoleWithSAML` non è firmata (crittografata). Pertanto, è opportuno includere le policy di sessione facoltative solo se la richiesta viene trasmessa attraverso un intermediario affidabile. In questo caso, qualcuno potrebbe modificare la policy per rimuovere le limitazioni.

Si tratta di una chiamata non firmata, che significa che l'app non ha bisogno di accedere a qualsiasi credenziali di sicurezza AWS per effettuare la chiamata. Quando si effettua questa chiamata, vengono fornite le seguenti informazioni:

- L'Amazon Resource Name (ARN) del ruolo che deve assumere l'app.
- L'ARN del provider SAML creato in IAM che descrive il provider di identità.
- L'asserzione SAML, codificata in base 64, che è stata fornita dal provider di identità SAML nella risposta di autenticazione alla richiesta di accesso proveniente dall'app.

- È possibile configurare il proprio IdP per passare attributi nella propria asserzione SAML come tag di sessione.
- (Facoltativo) Durata, che specifica la durata delle credenziali di sicurezza temporanee. Utilizzare il parametro `DurationSeconds` per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo. Se non si passa questo parametro, le credenziali temporanee scadono in un'ora. Il parametro `DurationSeconds` da questa API è separato dal parametro HTTP `SessionDuration` che viene utilizzato per specificare la durata di una sessione di console. Utilizzare il parametro HTTP `SessionDuration` nella richiesta all'endpoint di federazione per un token di accesso alla console. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.
- (Facoltativo) Policy di gestione inline o gestite. Queste policy limitano le autorizzazioni derivate dalla policy basata sull'identità del ruolo che sono assegnate alla sessione del ruolo. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.
- Nome della sessione del ruolo Utilizza questo valore stringa per identificare la sessione quando un ruolo viene utilizzato da principali diversi. Per motivi di sicurezza, gli amministratori possono visualizzare questo campo nei log AWS CloudTrail per sapere chi ha eseguito un'operazione in AWS. L'amministratore potrebbe richiedere di fornire un valore specifico per il nome della sessione quando si assume il ruolo. Per ulteriori informazioni, consulta `sts:RoleSessionName`.
- (Facoltativo) Identità di origine. Puoi richiedere agli utenti federati di specificare un'identità di origine quando assumono un ruolo. Dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nella richiesta di tutte le operazioni intraprese durante la sessione del ruolo. Il valore dell'identità di origine persiste tra le varie sessioni di ruolo concatenati. Puoi utilizzare le informazioni di identità di origine nei AWS CloudTrail log per determinare chi ha intrapreso azioni con un ruolo. Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Quando chiami`AssumeRoleWithSAML`, AWS verifica l'autenticità dell'asserzione SAML. Supponendo che il provider di identità convalidi l'asserzione, AWS ti restituisce le seguenti informazioni:

- Un set di credenziali di sicurezza temporanee. Consistono in un ID chiave di accesso, in una Secret Access Key e in un token di sessione.
- l'ID del ruolo e l'ARN del ruolo assunto.
- Un valore `Audience` che contiene il valore dell'attributo `Recipient` dell'elemento `SubjectConfirmationData` dell'asserzione SAML.
- Un valore `Issuer` che contiene il valore dell'attributo `Issuer` dell'elemento dell'asserzione SAML.
- Un `NameQualifier` elemento che contiene un valore hash costruito a partire dal `Issuer` valore, dall' Account AWS ID e dal nome descrittivo del provider SAML. Quando combinato con l'elemento `Subject`, possono identificare in modo univoco l'utente federato.
- Un elemento `Subject` che contiene il valore dell'elemento `NameID` nell'elemento `Subject` dell'asserzione SAML.
- Un elemento `SubjectType` che indica il formato dell'elemento `Subject`. Il valore può essere `persistent`, `transient` o il pieno `Format` URI dagli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

Se disponi delle credenziali di sicurezza temporanee, puoi utilizzarle per effettuare AWS chiamate API. Si tratta dello stesso processo utilizzato per effettuare una chiamata AWS API con credenziali di sicurezza a lungo termine. La differenza è che è necessario includere il token della sessione, che consente ad AWS di verificare che le credenziali di sicurezza provvisorie siano valide.

L'app deve memorizzare le credenziali. Come impostazione predefinita, le credenziali scadono dopo un'ora. Se non utilizzi l'CredentialsProviderazione AmazonSTS nell' AWS SDK, spetta a te e alla tua app effettuare nuovamente la chiamata. `AssumeRoleWithSAML` Chiamare questa operazione per ottenere un nuovo set di credenziali di sicurezza provvisorie prima che i vecchi scadere.

GetFederationToken: federazione tramite un gestore identità personalizzato

L'operazione API `GetFederationToken` restituisce un set di credenziali di sicurezza temporanee per gli utenti federati. Questa API differisce da `AssumeRole` per il fatto che il periodo di scadenza predefinito è notevolmente più lungo (12 ore invece di un'ora). Inoltre, è possibile utilizzare il parametro `DurationSeconds` per specificare una durata per le credenziali di sicurezza temporanee perché rimanga valido. Le credenziali risultanti sono valide per la durata specificata, compresa tra 900 secondi (15 minuti) e 129.600 secondi (36 ore). Il periodo di scadenza più lungo può aiutare a ridurre il numero di chiamate, in AWS quanto non è necessario ottenere nuove credenziali con la stessa frequenza.

Quando effettui questa richiesta, vengono utilizzate le credenziali di un utente IAM specifico. Le autorizzazioni per le credenziali di sicurezza temporanee sono determinate dalle policy di sessione passate al momento della chiamata di `GetFederationToken`. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy utente IAM e le policy di sessione fornite. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalle policy basate sull'identità dell'utente IAM che richiede la federazione. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

Quando si utilizzano le credenziali temporanee restituite dall'operazione `GetFederationToken`, i tag di entità della sessione includono i tag dell'utente e i tag di sessione passati. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.

La chiamata `GetFederationToken` restituisce le credenziali di sicurezza temporanee che consistono nel token di sessione, nella chiave di accesso, nella chiave segreta e nella scadenza. È possibile utilizzare `GetFederationToken` se si desidera gestire le autorizzazioni nella propria organizzazione (ad esempio, utilizzando l'applicazione proxy per assegnare le autorizzazioni).

L'esempio seguente mostra una richiesta di esempio e la risposta che utilizza `GetFederationToken`. Questa richiesta di esempio consente di federare l'utente chiamante per la durata specificata tramite l'ARN della policy di sessione e i tag di sessione. La sessione risultante è denominata `Jane-session`.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetFederationToken
&Name=Jane-session
&PolicyArns.member.1.arn==arn%3Aaws%3Aiam%3A%3A123456789012%3Apolicy%2FRole1policy
&DurationSeconds=1800
&Tags.member.1.Key=Project
&Tags.member.1.Value=Pegasus
&Tags.member.2.Key=Cost-Center
&Tags.member.2.Value=12345
&AUTHPARAMS
```

L'ARN della policy illustrato nell'esempio precedente include i seguenti ARN URL-encoded:

`arn:aws:iam::123456789012:policy/Role1policy`

Inoltre, si noti che il parametro &AUTHPARAMS nell'esempio è inteso come segnaposto per le informazioni di autenticazione. Questa è la firma, che devi includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vai a Firmare AWS le richieste utilizzando la versione 4 della pagina Riferimenti generali di Amazon Web Servicesper scoprire come firmare una richiesta.

Oltre alle credenziali di sicurezza temporanee, la risposta include l'Amazon Resource Name (ARN) per l'utente federato e il periodo di scadenza delle credenziali.

Example Example response

```
<GetFederationTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetFederationTokenResult>
<Credentials>
  <SessionToken>
   AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
   LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
   QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
   9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
   +scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCEXAMPLE==
  </SessionToken>
  <SecretAccessKey>
  wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2019-04-15T23:28:33.359Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE;</AccessKeyId>
</Credentials>
<FederatedUser>
  <Arn>arn:aws:sts::123456789012:federated-user/Jean</Arn>
  <FederatedUserId>123456789012:Jean</FederatedUserId>
</FederatedUser>
<PackedPolicySize>4</PackedPolicySize>
</GetFederationTokenResult>
<ResponseMetadata>
<RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
</ResponseMetadata>
</GetFederationTokenResponse>
```

> **Note**
>
> Una AWS conversione comprime le politiche di sessione e i tag di sessione passati in un formato binario compresso con un limite separato. La richiesta può non essere eseguita correttamente a causa di questo limite anche se il testo in chiaro soddisfa gli altri requisiti. L'elemento della risposta `PackedPolicySize` indica in percentuale la consistenza di policy e tag della richiesta rispetto al limite di dimensione superiore.

AWS consiglia di concedere le autorizzazioni a livello di risorsa (ad esempio, si allega una policy basata sulle risorse a un bucket Amazon S3), è possibile omettere il parametro. `Policy` Tuttavia, se non si include una policy per l'utente federato, le credenziali di sicurezza temporanee non concederanno le autorizzazioni. In questo caso, è necessario utilizzare le policy delle risorse per concedere all'utente federato l'accesso alle risorse AWS .

Ad esempio, supponiamo che il tuo Account AWS numero sia 111122223333 e che tu disponga di un bucket Amazon S3 a cui desideri consentire l'accesso a Susan. Le credenziali di sicurezza temporanee di Susan non includono una policy per il bucket. In questo caso, è necessario accertarsi che il bucket disponga di una policy con un ARN corrisponda all'ARN di Susan, ad esempio `arn:aws:sts::111122223333:federated-user/Susan`.

GetSessionToken: le credenziali temporanee per gli utenti in ambienti non attendibili

L'operazione API `GetSessionToken` restituisce un set di credenziali di sicurezza temporanee a un utente IAM esistente. Ciò è utile per fornire una maggiore sicurezza, ad esempio consentire le AWS richieste solo quando l'MFA è abilitata per l'utente IAM. Poiché le credenziali sono temporanee, forniscono maggiore sicurezza quando hai un utente IAM che accede alle risorse tramite un ambiente meno sicuro. Esempi di ambienti meno protetti includono dispositivi mobili o browser web. Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee o GetSessionTokenconsulta l'AWS Security Token Service API Reference.

Per impostazione predefinita, le credenziali di sicurezza temporanee per un utente IAM sono valide per un massimo di 12 ore. Tuttavia, è possibile richiedere una durata di soli 15 minuti o fino a 36 ore utilizzando il parametro `DurationSeconds`. Per motivi di sicurezza, un token per un Utente root dell'account AWS è limitato a una durata di un'ora.

`GetSessionToken` restituisce le credenziali di sicurezza temporanee, ossia un token di sessione, l'ID chiave di accesso e una chiave di accesso segreta. L'esempio seguente mostra una richiesta

di esempio e la risposta utilizzando `GetSessionToken`. La risposta include inoltre il periodo di scadenza delle credenziali di sicurezza temporanee.

Example Richiesta di esempio

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetSessionToken
&DurationSeconds=1800
&AUTHPARAMS
```

Il parametro AUTHPARAMS nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che è necessario includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, vai a Firmare AWS le richieste utilizzando la versione 4 della pagina Riferimenti generali di Amazon Web Servicesper scoprire come firmare una richiesta.

Example Example response

```
<GetSessionTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetSessionTokenResult>
<Credentials>
  <SessionToken>
   AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT+FvwqnKwRcOIfrRh3c/L
   To6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/IvU1dYUg2RVAJBanLiHb4IgRmpRV3z
   rkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/AXlzBBko7b15fjrBs2+cTQtp
   Z3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE
  </SessionToken>
  <SecretAccessKey>
  wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
  </SecretAccessKey>
  <Expiration>2011-07-11T19:55:29.611Z</Expiration>
  <AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
</GetSessionTokenResult>
<ResponseMetadata>
<RequestId>58c5dbae-abef-11e0-8cfe-09039844ac7d</RequestId>
</ResponseMetadata>
</GetSessionTokenResponse>
```

Facoltativamente, la `GetSessionToken` richiesta può includere `TokenCode` valori per `SerialNumber` la verifica dell' AWS autenticazione a più fattori (MFA). Se i valori forniti sono validi, AWS STS fornisce credenziali di sicurezza temporanee che includono lo stato dell'autenticazione MFA. Le credenziali di sicurezza temporanee possono quindi essere utilizzate per accedere alle operazioni o ai AWS siti Web dell'API protetti da MFA finché l'autenticazione MFA è valida.

L'esempio seguente mostra una richiesta `GetSessionToken`, che include un codice di verifica MFA e il numero di serie del dispositivo.

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetSessionToken
&DurationSeconds=7200
&SerialNumber=YourMFADeviceSerialNumber
&TokenCode=123456
&AUTHPARAMS
```

> **Note**
>
> La chiamata a AWS STS può essere diretta all'endpoint globale o a uno qualsiasi degli endpoint regionali su cui attivi. Account AWS Per ulteriori informazioni, consulta la sezione di AWS STS relativa alle regioni e agli endpoint.
> Il parametro `AUTHPARAMS` nell'esempio è un segnaposto per la propria firma. Una firma è l'informazione di autenticazione che devi includere nelle richieste API AWS HTTP. Per creare le richieste API ti consigliamo di utilizzare gli SDK AWS poiché, tra gli altri vantaggi, gli SDK gestiscono la firma delle richieste per tuo conto. Se devi creare e firmare le richieste API manualmente, consulta Firmare AWS le richieste utilizzando la versione 4 della Riferimenti generali di Amazon Web Servicespagina per scoprire come firmare una richiesta.

Confronto delle operazioni AWS STS API

La tabella seguente confronta le funzionalità delle operazioni API AWS STS che restituiscono credenziali di sicurezza temporanee. Per informazioni sui diversi metodi che si possono utilizzare per richiedere le credenziali di sicurezza temporanee assumendo un ruolo, consultare Utilizzo di ruoli IAM. Per ulteriori informazioni sulle diverse operazioni AWS STS API che consentono di passare i tag di sessione, consulta. Passare i tag di sessione AWS STS

Confronto tra le opzioni API

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
AssumeRole	Utente IAM o ruolo IAM con credenziali di sicurezza temporanee esistenti	15 min | Impostazione durata massima della sessione[3] | 1 ora	Sì	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .
AssumeRoleWithSAML	Qualsiasi intermediario utente deve passare una risposta di autenticazione SAML che indica l'autenticazione da un provider di identità noto	15 min | Impostazione durata massima della sessione[3] | 1 ora	No	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .
AssumeRoleWithWebIdentity	Qualsiasi utente; il chiamante deve passare un token JWT conforme a OIDC che indichi	15 min | Impostazione durata massima	No	Sì	Non è possibile chiamare `GetFederationToken` o `GetSessionToken` .

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
	l'autenticazione da un provider di identità noto	della sessione[3] | 1 ora			
GetFederationToken	Utente IAM o Utente root dell'account AWS	Utente IAM: 15 m | 36 ore | 12 ore Utente root: 15 m | 1 ora | 1 ora	No	Sì	Impossibile chiamare le operazioni IAM utilizzando l' AWS API AWS CLI o. Questa limitazione non si applica alle sessioni della console. Impossibile chiamare AWS STS le operazioni ad eccezione di `GetCallerIdentity` .4 L'accesso SSO alla console è consentito.[5]

AWS STS API	Chi può chiamare	Ciclo di vita delle credenziali (minimo | massimo | predefinito)	Supporto MFA[1]	Supporto di policy di sessione[2]	Restrizioni per le credenziali provvisorie risultanti
GetSessionToken	Utente IAM o Utente root dell'account AWS	Utente IAM: 15 m | 36 ore | 12 ore Utente root: 15 m | 1 ora | 1 ora	Sì	No	Impossibile chiamare le operazioni API IAM a meno che le informazioni di MFA siano incluse con la richiesta. Impossibile chiamare le operazioni AWS STS API tranne `AssumeRole` o`GetCallerIdentity` . L'accesso SSO alla console non è consentito.[6]

[1] Supporto MFA. Puoi includere informazioni su un dispositivo di autenticazione a più fattori (MFA) quando chiami AssumeRole le operazioni GetSessionToken e API. In questo modo le credenziali di sicurezza provvisorie risultanti dalla chiamata API possono essere utilizzate solo dagli utenti autenticati con un dispositivo MFA. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto da MFA.

[2] Supporto per la policy di sessione. Le policy di sessione sono policy che si passano come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Questa policy limita le autorizzazioni dalla policy basata su identità del ruolo o dell'utente che sono assegnate alla sessione. Le autorizzazioni della sessione risultanti sono l'intersezione delle policy basate sull'identità dell'entità e delle policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata

sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

[3] Impostazione della durata massima della sessione. Utilizzare il parametro `DurationSeconds` per specificare la durata della sessione del ruolo da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.

3. GetCallerIdentity Non sono necessarie autorizzazioni per eseguire questa operazione. Se un amministratore aggiunge una policy al tuo utente o ruolo IAM che nega esplicitamente l'accesso all'operazione `sts:GetCallerIdentity`, puoi comunque eseguire questa operazione. Le autorizzazioni non sono necessarie perché le stesse informazioni vengono restituite quando a un utente o ruolo IAM viene negato l'accesso. Per visualizzare un esempio di risposta, consulta Non sono autorizzato a eseguire: iam: MFADevice DeleteVirtual.

[5] Accesso Single Sign-On (SSO) alla console. Per supportare l'SSO, AWS consente di chiamare un endpoint di federazione (`https://signin.aws.amazon.com/federation`) e passare credenziali di sicurezza temporanee. L'endpoint restituisce un token che è possibile utilizzare per creare un URL che effettua l'accesso di un utente direttamente nella console senza richiedere una password. Per ulteriori informazioni, consulta Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console e Come abilitare l'accesso tra più account alla console di AWS gestione nel blog sulla sicurezza. AWS

2 Dopo aver recuperato le credenziali temporanee, non è possibile accedervi passando le credenziali all' AWS Management Console endpoint Single Sign-On della federazione. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

Utilizzo di credenziali temporanee con le risorse AWS

Puoi utilizzare credenziali di sicurezza temporanee per effettuare richieste programmatiche di AWS risorse utilizzando l' AWS API AWS CLI or (utilizzando gli SDK).AWS Le credenziali temporanee forniscono le stesse autorizzazioni delle credenziali di sicurezza a lungo termine, come ad esempio le credenziali degli utenti IAM. Tuttavia, ci sono alcune differenze:

- Quando si effettua una chiamata utilizzando credenziali di sicurezza temporanee, la chiamata deve includere un token di sessione, che viene restituito insieme a tali credenziali temporanee. AWS utilizza il token di sessione per convalidare le credenziali di sicurezza temporanee.
- Le credenziali temporanee scadono dopo un intervallo di tempo specificato. Dopo che le credenziali temporanee scadono, tutte le chiamate effettuate con tali credenziali verranno respinte,

pertanto dovrai generare un nuovo set di credenziali temporanee. Le credenziali temporanee non possono essere prorogate o aggiornate oltre l'intervallo specificato in origine.

- Quando si utilizzano credenziali temporanee per effettuare una richiesta, l'entità potrebbe includere un set di tag. Questi tag provengono da tag di sessione e tag associati al ruolo assunto. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.

Se utilizzi gli AWS SDK, il AWS Command Line Interface(AWS CLI) o gli strumenti per Windows PowerShell, il modo per ottenere e utilizzare le credenziali di sicurezza temporanee varia a seconda del contesto. Se esegui codice o PowerShell comandi Tools for Windows all'interno di un'istanza EC2, puoi sfruttare i ruoli per Amazon EC2. AWS CLI Altrimenti, puoi richiamare un'API AWS STS per ottenere le credenziali provvisorie e utilizzarle in modo esplicito per effettuare chiamate ai servizi AWS .

> **Note**
>
> Puoi usare AWS Security Token Service (AWS STS) per creare e fornire a utenti affidabili credenziali di sicurezza temporanee in grado di controllare l'accesso alle tue risorse. AWS Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. AWS STS è un servizio globale con un endpoint predefinito in`https://sts.amazonaws.com`. Questo endpoint si trova nella regione Stati Uniti orientali (Virginia settentrionale), sebbene le credenziali ottenute da questo e da altri endpoint siano valide a livello globale. Queste credenziali funzionano con servizi e risorse in qualsiasi regione. Puoi anche scegliere di effettuare chiamate AWS STS API verso gli endpoint in una qualsiasi delle regioni supportate. Ciò può ridurre la latenza effettuando le richieste da server in una regione geograficamente più vicina a te. Indipendentemente dalla regione dalla quale provengono, le credenziali funzionano a livello globale. Per ulteriori informazioni, consulta Gestire AWS STS in un Regione AWS.

Indice

Utilizzo delle credenziali temporanee nelle istanze Amazon EC2

Se desideri eseguire AWS CLI comandi o codice all'interno di un'istanza EC2, il metodo consigliato per ottenere le credenziali consiste nell'utilizzare i ruoli per Amazon EC2. È possibile creare un ruolo IAM che specifichi le autorizzazioni che si desidera concedere alle applicazioni che vengono eseguite sulle istanze EC2. Quando si avvia l'istanza, si associa il ruolo all'istanza.

Le applicazioni e AWS CLI i PowerShell comandi Tools for Windows eseguiti sull'istanza possono quindi ottenere credenziali di sicurezza temporanee automatiche dai metadati dell'istanza. Non è necessario ottenere esplicitamente le credenziali di sicurezza temporanee. Gli AWS SDK e gli strumenti per Windows ottengono PowerShell automaticamente le credenziali dall'EC2 Instance Metadata Service (IMDS) e le utilizzano. AWS CLI Le credenziali temporanee hanno le autorizzazioni che si definiscono per il ruolo associato all'istanza.

Per maggiori informazioni ed esempi, consultare quanto segue:

- Utilizzo dei ruoli IAM per concedere l'accesso alle AWS risorse su Amazon Elastic Compute Cloud — AWS SDK for Java
- Concessione dell'accesso utilizzando un ruolo IAM — AWS SDK for .NET
- Creazione di un ruolo: AWS SDK for Ruby

Utilizzo delle credenziali di sicurezza temporanee con gli SDK AWS

Per utilizzare credenziali di sicurezza temporanee nel codice, chiamate a livello di codice un' AWS STS API simile a un'API `AssumeRole` ed estraete le credenziali e il token di sessione risultanti. Questi valori vengono quindi utilizzati come credenziali per le chiamate successive a. AWS L'esempio seguente mostra lo pseudocodice su come utilizzare le credenziali di sicurezza temporanee se utilizzi un SDK: AWS

```
assumeRoleResult = AssumeRole(role-arn);
tempCredentials = new SessionAWSCredentials(
   assumeRoleResult.AccessKeyId,
   assumeRoleResult.SecretAccessKey,
   assumeRoleResult.SessionToken);
s3Request = CreateAmazonS3Client(tempCredentials);
```

Per un esempio scritto in Python (usando AWS SDK for Python (Boto)), consultare Passaggio a un ruolo IAM (AWS API). In questo esempio viene illustrato come richiamare `AssumeRole` per ottenere

le credenziali di sicurezza temporanee e quindi utilizzare tali credenziali per effettuare una chiamata ad Amazon S3.

Per dettagli su come richiamare `AssumeRole`, `GetFederationToken` e altre operazioni API, consulta la Documentazione di riferimento delle API AWS Security Token Service. Per informazioni su come ottenere le credenziali di sicurezza provvisorie e il token di sessione dal risultato, consulta la documentazione dell'SDK in uso. Puoi trovare la documentazione per tutti gli AWS SDK nella pagina di AWS documentazione principale, nella sezione SDK e Toolkit.

È necessario accertarsi che sia possibile ottenere un nuovo set di credenziali prima della scadenza. In alcuni SDK, è possibile utilizzare un provider che gestisca il proprio processo di aggiornamento delle credenziali; controllare la documentazione del kit SDK che si sta utilizzando.

Utilizzo delle credenziali di sicurezza temporanee con la AWS CLI

È possibile utilizzare le credenziali di sicurezza temporanee con AWS CLI Questo può essere utile per testare le policy.

Tramite la AWS CLI, puoi richiamare un'API AWS STS come `AssumeRole` o `GetFederationToken` e acquisire l'output risultante. L'esempio seguente mostra una chiamata a `AssumeRole` che invia l'output a un file. Nell'esempio, si presume che il `profile` parametro sia un profilo nel file di configurazione. AWS CLI Si presume inoltre di fare riferimento alle credenziali di un utente IAM che disponga delle autorizzazioni per assumere il ruolo.

```
aws sts assume-role --role-arn arn:aws:iam::123456789012:role/role-name --role-session-name "RoleSession1" --profile IAM-user-name > assume-role-output.txt
```

Quando il comando viene completato, è possibile estrarre l'ID della chiave di accesso, la chiave di accesso segreta e il token di sessione da qualunque posto sia stato instradato. È possibile farlo manualmente o utilizzando uno script. È possibile assegnare questi valori alle variabili di ambiente.

Quando AWS CLI esegui i comandi, AWS CLI cerca le credenziali in un ordine specifico, prima nelle variabili di ambiente e poi nel file di configurazione. Pertanto, dopo aver inserito le credenziali temporanee nelle variabili di ambiente, AWS CLI utilizza tali credenziali per impostazione predefinita. (Se specificate un `profile` parametro nel comando, AWS CLI salta le variabili di ambiente. Al contrario, viene AWS CLI visualizzato nel file di configurazione, che consente di sovrascrivere le credenziali nelle variabili di ambiente, se necessario.)

L'esempio seguente mostra come impostare le variabili di ambiente per le credenziali di sicurezza temporanee e quindi chiamare un comando. AWS CLI Poiché nel AWS CLI comando non è incluso

alcun `profile` parametro, AWS CLI cerca le credenziali prima nelle variabili di ambiente e quindi utilizza le credenziali temporanee.

Linux

```
$ export AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
$ aws ec2 describe-instances --region us-west-1
```

Windows

```
C:\> SET AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
C:\> SET AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> SET AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of token>
C:\> aws ec2 describe-instances --region us-west-1
```

Utilizzo delle credenziali di sicurezza temporanee con le operazioni API

Se stai effettuando richieste API HTTPS dirette a AWS, puoi firmare tali richieste con le credenziali di sicurezza temporanee che ottieni da (). AWS Security Token Service AWS STS A tale scopo, utilizzate l'ID della chiave di accesso e la chiave di accesso segreta da AWS STS cui ricevete. Utilizzare l'ID della chiave di accesso e la chiave di accesso segreta nello stesso modo in cui si utilizzano le credenziali a lungo termine per firmare una richiesta. Inoltre, aggiungi alla tua richiesta API il token di sessione da cui ricevi AWS STS. Aggiungere il token della sessione a un'intestazione HTTP o a un parametro della stringa di query denominato `X-Amz-Security-Token`. Aggiungi il token di sessione all'intestazione HTTP o il parametro della stringa di query, ma non entrambi. Per ulteriori informazioni sulla firma delle richieste API HTTPS, consulta Firmare le richieste AWS API in Riferimenti generali di AWS.

Ulteriori informazioni

Per ulteriori informazioni sull'utilizzo AWS STS con altri AWS servizi, consulta i seguenti collegamenti:

- Amazon S3. Consulta Esecuzione di richieste mediante le credenziali temporanee per gli utenti IAM o Esecuzione di richieste mediante le credenziali temporanee per gli utenti federati nella Guida per l'utente di Amazon Simple Storage Service.
- Amazon SNS Vedi Utilizzo di politiche basate sull'identità con Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

- Amazon SQS Consulta la gestione delle identità e degli accessi in Amazon SQS nella Amazon Simple Queue Service Developer Guide.
- Amazon SimpleDB Consulta Utilizzo di credenziali di sicurezza temporanee nella Guida per gli sviluppatori di Amazon SimpleDB.

Controllo delle autorizzazioni per le credenziali di sicurezza temporanee

È possibile utilizzare AWS Security Token Service (AWS STS) per creare e fornire agli utenti attendibili credenziali di sicurezza temporanee in grado di controllare l'accesso alle risorse. AWS Per ulteriori informazioni su AWS STS, vedereCredenziali di sicurezza temporanee in IAM. Le credenziali di sicurezza temporanee emesse da AWS STS sono valide fino al periodo di scadenza e non possono essere revocate. Tuttavia, le autorizzazioni assegnate a tali credenziali vengono valutate ogni volta che viene effettuata una richiesta che utilizza le credenziali stesse, pertanto è possibile ottenere l'effetto di revoca delle credenziali modificando i relativi diritti di accesso dopo che sono state emesse.

Negli argomenti seguenti si presuppone che tu abbia una conoscenza pratica delle AWS autorizzazioni e delle politiche. Per ulteriori informazioni su questi argomenti, consultare Gestione degli accessi AWS alle risorse.

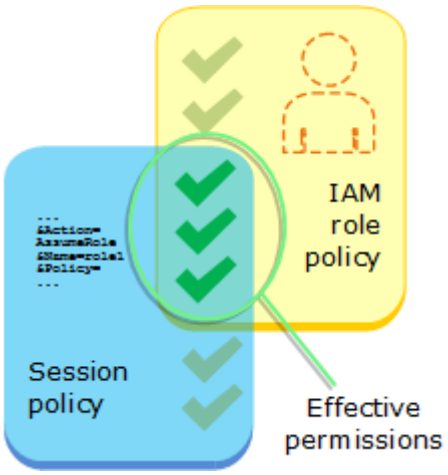
Argomenti

- Autorizzazioni per AssumeRole, AssumeRoleWith SAML e AssumeRoleWithWebIdentity
- Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti
- Autorizzazioni per GetFederationToken
- Autorizzazioni per GetSessionToken
- Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee
- Concessione delle autorizzazioni per creare credenziali di sicurezza temporanee
- Concessione delle autorizzazioni per l'utilizzo di sessioni di console con riconoscimento dell'identità

Autorizzazioni per AssumeRole, AssumeRoleWith SAML e AssumeRoleWithWebIdentity

La policy di autorizzazione del ruolo assunto determina le autorizzazioni per le credenziali di sicurezza temporanee restituite da `AssumeRole`, `AssumeRoleWithSAML` e `AssumeRoleWithWebIdentity`. Puoi definire queste autorizzazioni quando crei o aggiorni il ruolo.

Facoltativamente, puoi trasferire le policy di sessione inline o gestite come parametri delle operazioni API `AssumeRole`, `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity`. Le policy di sessione limitano le autorizzazioni per la sessione con credenziali temporanee del ruolo. Le autorizzazioni della sessione risultante sono l'intersezione della policy basata sull'identità del ruolo e delle policy di sessione. Puoi utilizzare le credenziali temporanee del ruolo nelle successive chiamate AWS API per accedere alle risorse dell'account proprietario del ruolo. Non puoi utilizzare policy di sessione per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata su identità del ruolo che viene assunto. Per ulteriori informazioni su come AWS determina le autorizzazioni valide di un ruolo, consulta Logica di valutazione delle policy.



Le politiche allegate alle credenziali a cui è stata effettuata la chiamata originale non `AssumeRole` vengono valutate AWS quando si prende la decisione di autorizzazione «consentire» o «negare». L'utente rinuncia temporaneamente alle autorizzazioni originali a favore delle autorizzazioni assegnate dal ruolo assunto. Nel caso delle operazioni `AssumeRoleWithSAML` e dell'`AssumeRoleWithWebIdentity`API, non ci sono policy da valutare perché il chiamante dell'API non è un'identità. AWS

Esempio: assegnazione di autorizzazioni utilizzando AssumeRole

È possibile usare l'operazione API `AssumeRole` con diversi tipi di policy. Di seguito sono illustrati alcuni esempi.

Policy di autorizzazione di un ruolo

In questo esempio chiami l'operazione API `AssumeRole` senza specificare la policy di sessione nel parametro `Policy` facoltativo. Le autorizzazioni assegnate alle credenziali temporanee sono determinate dalla policy di autorizzazione del ruolo assunto. La policy di autorizzazioni di esempio seguente concede al ruolo l'autorizzazione per elencare tutti gli oggetti contenuti in un bucket S3

denominato `productionapp`. Consente inoltre al ruolo di ottenere, inserire ed eliminare gli oggetti all'interno del bucket.

Example Policy di autorizzazione di un ruolo di esempio

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::productionapp"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::productionapp/*"
    }
  ]
}
```

Policy di sessione passata come parametro

Immaginiamo di voler consentire a un utente di assumere lo stesso ruolo dell'esempio precedente. In questo caso però il ruolo della sessione deve avere l'autorizzazione solo per ottenere e mettere oggetti nel bucket S3 `productionapp`. Non desideri permettere all'utente di eliminare gli oggetti. Un metodo per raggiungere questo scopo consiste nel creare un nuovo ruolo e specificare le autorizzazioni desiderate nella policy di autorizzazione di tale ruolo. Un altro metodo per raggiungere lo scopo consiste nel chiamare l'API `AssumeRole` e includere una policy di sessione nel parametro `Policy` facoltativo come parte dell'operazione API. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. Le policy di sessione non possono essere utilizzate per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata sull'identità del ruolo che viene assunto. Per ulteriori informazioni sulle autorizzazioni della sessione del ruolo, consulta Policy di sessione.

Dopo aver recuperato le credenziali temporanee della nuova sessione, puoi passarle all'utente che deve disporre di tali autorizzazioni.

Immagina, ad esempio, che la policy seguente venga passata come parametro della chiamata API. L'utente che utilizza la sessione dispone di autorizzazioni per eseguire solo le seguenti azioni:

- Elencare tutti gli oggetti nel bucket `productionapp`.
- Ottenere e inserire gli oggetti nel bucket `productionapp`.

Nella policy di sessione seguente, l'autorizzazione `s3:DeleteObject` viene esclusa e alla sessione assunta non viene concessa l'autorizzazione `s3:DeleteObject`. La policy imposta il numero massimo di autorizzazioni per la sessione del ruolo, in modo che sostituisca qualsiasi policy di autorizzazione esistente su quel ruolo.

Example Esempio di policy di sessione passata con la chiamata API **`AssumeRole`**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::productionapp"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::productionapp/*"
    }
  ]
}
```

Policy basata su risorse

Alcune AWS risorse supportano politiche basate sulle risorse e queste politiche forniscono un altro meccanismo per definire le autorizzazioni che influiscono sulle credenziali di sicurezza temporanee. Solo alcune risorse, come i bucket Amazon S3, gli argomenti Amazon SNS e le code Amazon SQS, supportano le policy basate sulle risorse. L'esempio seguente fornisce ulteriori informazioni sugli esempi precedenti, utilizzando un bucket S3, denominato `productionapp`. La policy seguente è collegata al bucket.

Quando colleghi la seguente policy basata su risorse al bucket `productionapp`, a tutti gli utenti viene negata l'autorizzazione per eliminare gli oggetti dal bucket. (Consulta l'elemento `Principal` nella policy). Ciò include tutti gli utenti che assumono il ruolo, anche se la policy di autorizzazione del ruolo concede l'autorizzazione `DeleteObject`. Un'istruzione `Deny` esplicita ha sempre la precedenza su un'istruzione `Allow`.

Example Esempio di policy di bucket

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": {"AWS": "*"},
    "Effect": "Deny",
    "Action": "s3:DeleteObject",
    "Resource": "arn:aws:s3:::productionapp/*"
  }
}
```

Per ulteriori informazioni su come più tipi di policy vengono combinati e valutati da. AWSLogica di valutazione delle policy

Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti

Un ruolo IAM è un oggetto in IAM a cui sono assegnate delle autorizzazioni. Quando assumi quel ruolo utilizzando un'identità IAM o un'identità esterna AWS, ricevi una sessione con le autorizzazioni assegnate al ruolo.

Quando esegui azioni in AWS, le informazioni sulla sessione possono essere registrate AWS CloudTrail per essere monitorate dall'amministratore dell'account. Gli amministratori possono configurare i ruoli in modo da richiedere alle identità di inviare una stringa personalizzata che identifichi la persona o l'applicazione che esegue operazioni in AWS. Queste informazioni di identità vengono archiviate come identità di origine in AWS CloudTrail. Quando l'amministratore esamina l'attività in CloudTrail, può visualizzare le informazioni sull'identità di origine per determinare chi o cosa ha eseguito le azioni nelle sessioni di ruolo assunte.

Una volta impostata, l'identità di origine è presente nelle richieste di qualsiasi AWS azione intrapresa durante la sessione di ruolo. Il valore impostato persiste quando un ruolo viene utilizzato per assumere un altro ruolo tramite l' AWS API AWS CLI o, nota come concatenamento dei ruoli.
Il valore impostato non può essere modificato durante la sessione del ruolo. Gli amministratori

possono configurare autorizzazioni granulari in base alla presenza o al valore dell'identità di origine per controllare ulteriormente AWS le azioni intraprese con ruoli condivisi. È possibile decidere se l'attributo dell'identità di origine può essere utilizzato, se è obbligatorio e quale valore può essere utilizzato.

Il modo in cui si utilizza l'identità di origine differisce dal nome della sessione di ruolo e dai tag di sessione in modo importante. Il valore dell'identità di origine non può essere modificato dopo l'impostazione e persiste per eventuali operazioni aggiuntive eseguite con la sessione del ruolo. Ecco come utilizzare i tag di sessione e il nome della sessione del ruolo:

- Tag di sessione: puoi passare i tag di sessione quando assumi un ruolo o federi un utente. I tag di sessione sono presenti quando si assume un ruolo. È possibile definire policy che utilizzano le chiavi condizionali sui tag per concedere autorizzazioni ai principali sulla base dei relativi tag. Quindi puoi utilizzarle CloudTrail per visualizzare le richieste fatte per assumere ruoli o federare gli utenti. Per ulteriori informazioni sui tag di sessione, consulta Passare i tag di sessione AWS STS.
- Nome sessione del ruolo: puoi utilizzare la chiave di condizione `sts:RoleSessionName` in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Il nome della sessione del ruolo può essere utilizzato per distinguere le sessioni del ruolo quando un ruolo viene utilizzato da principali diversi. Per saperne di più sul nome della sessione di ruolo, vedi sts: RoleSessionName.

Si consiglia di utilizzare l'identità di origine quando si desidera controllare l'identità che assume un ruolo. L'identità di origine è utile anche per CloudTrail i log di mining per determinare chi ha utilizzato il ruolo per eseguire azioni.

Argomenti

- Configurazione per l'uso dell'identità di origine
- Cose da sapere sull'identità di origine
- Autorizzazioni necessarie per impostare l'identità di origine
- Specifica di un'identità di origine quando si assume un ruolo
- Utilizzo dell'identità di origine con AssumeRole
- Utilizzo dell'identità di origine con SAML AssumeRoleWith
- Utilizzo dell'identità di origine con AssumeRoleWithWebIdentity
- Controllo dell'accesso tramite le informazioni sull'identità di origine

- Visualizzazione dell'identità di origine in CloudTrail

Configurazione per l'uso dell'identità di origine

Il modo in cui si imposta l'utilizzo dell'identità di origine dipende dal metodo utilizzato quando si assumono i ruoli. Ad esempio, gli utenti IAM potrebbero assumere ruoli direttamente utilizzando l'operazione `AssumeRole`. Se disponi di identità aziendali, note anche come identità della forza lavoro, queste potrebbero accedere alle tue risorse utilizzando. AWS `AssumeRoleWithSAML` Se gli utenti finali accedono alle tue applicazioni per dispositivi mobile o Web, potrebbero farlo utilizzando `AssumeRoleWithWebIdentity`. Di seguito è riportata una panoramica di alto livello del flusso di lavoro che consente di comprendere come impostare l'utilizzo delle informazioni sull'identità di origine nell'ambiente esistente.

1. Configurazione di utenti e ruoli di test: in un ambiente di preproduzione, configura utenti e ruoli di prova e configura le relative policy per consentire l'impostazione di un'identità di origine.

   Se utilizzi un provider di identità (IdP) per le identità federate, configura l'IdP per inviare un attributo utente a scelta per l'identità di origine nell'asserzione o nel token.
2. Assunzione del ruolo: verifica l'assunzione di ruoli e l'invio di un'identità di origine con gli utenti e i ruoli impostati per il test.
3. Revisione CloudTrail: rivedi le informazioni sull'identità di origine per i ruoli di test nei tuoi registri. CloudTrail
4. Formazione degli utenti: dopo aver eseguito il test nell'ambiente di preproduzione, assicurati che gli utenti sappiano come inviare le informazioni sull'identità di origine, se necessario. Imposta una scadenza per il momento in cui sarà richiesto agli utenti di fornire un'identità di origine nell'ambiente di produzione.
5. Configurazione delle policy di produzione: configura le policy per l'ambiente di produzione e quindi aggiungile agli utenti e ai ruoli di produzione.
6. Monitora l'attività: monitora l'attività del ruolo di produzione utilizzando i CloudTrail log.

Cose da sapere sull'identità di origine

Quando si utilizza un'identità di origine, tieni presente quanto segue.

- Le policy di attendibilità per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:SetSourceIdentity`. Per i ruoli che non dispongono di questa autorizzazione nella policy di attendibilità, l'operazione `AssumeRole*` avrà esito negativo.

Se non desideri aggiornare la policy di attendibilità del ruolo per ogni ruolo, puoi utilizzare un'istanza IdP separata per passare l'identità di origine. Quindi, aggiungere l'autorizzazione `sts:SetSourceIdentity` solo ai ruoli connessi all'IdP separato.

- Quando un'identità imposta un'identità di origine, la chiave `sts:SourceIdentity` è presente nella richiesta. Per le azioni successive intraprese durante la sessione di ruolo, la chiave `aws:SourceIdentity` è presente nella richiesta. AWS non controlla il valore dell'identità di origine nelle chiavi `sts:SourceIdentity` o `aws:SourceIdentity`. Se decidi di richiedere un'identità di origine, è necessario scegliere un attributo che sia fornito dagli utenti o dall'IdP. Per motivi di sicurezza, è necessario assicurarsi di poter controllare il modo in cui tali valori vengono forniti.
- Il valore dell'identità di origine deve contenere tra 2 e 64 caratteri, può contenere solo caratteri alfanumerici, caratteri di sottolineatura e i seguenti caratteri: . , + = @ - (trattino). Non è possibile utilizzare un valore che inizia con il testo **`aws:`**. Questo prefisso è riservato all'uso AWS interno.
- Le informazioni sull'identità di origine non vengono acquisite CloudTrail quando un AWS servizio o un ruolo collegato al servizio esegue un'azione per conto di un'identità federata o della forza lavoro.

> **Important**
>
> Non è possibile passare a un ruolo AWS Management Console che richiede l'impostazione di un'identità di origine al momento dell'assunzione del ruolo. Per assumere un ruolo di questo tipo, è possibile utilizzare l' AWS API AWS CLI or per richiamare l'`AssumeRole`operazione e specificare il parametro source identity.

Autorizzazioni necessarie per impostare l'identità di origine

Oltre a quella sull'operazione che corrisponde all'operazione API, è necessario disporre nella policy dell'autorizzazione per le seguenti operazioni:

```
sts:SetSourceIdentity
```

- Per specificare un'identità di origine, i principali (utenti e ruoli IAM) devono disporre delle autorizzazioni per`sts:SetSourceIdentity`. In qualità di amministratore, puoi configurarlo nella policy di attendibilità del ruolo e nella policy di autorizzazione del principale.
- Quando si assume un ruolo con un altro ruolo, secondo la funzione denominata concatenamento dei ruoli, le autorizzazioni per `sts:SetSourceIdentity` sono necessarie sia nella policy di

autorizzazione del principale che assume il ruolo sia nella policy di attendibilità del ruolo del ruolo di destinazione. In caso contrario, l'operazione di assunzione del ruolo avrà esito negativo.

- Quando si utilizza l'identità di origine, le policy di attendibilità dei ruoli per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:SetSourceIdentity`. L'operazione `AssumeRole*` avrà esito negativo per qualsiasi ruolo connesso a un IdP senza questa autorizzazione. Se non desideri aggiornare la policy di attendibilità del ruolo per ogni ruolo, puoi utilizzare un'istanza IdP separata per inviare l'identità di origine e aggiungere l'autorizzazione `sts:SetSourceIdentity` solo ai ruoli connessi all'IdP separato.
- Per impostare un'identità di origine oltre i limiti dell'account, è necessario includere l'autorizzazione `sts:SetSourceIdentity` in due posti. Deve trovarsi nella policy di autorizzazione del principale nell'account di origine e nella policy di attendibilità del ruolo del ruolo nell'account di destinazione. Questa operazione potrebbe essere necessaria, ad esempio, quando un ruolo viene utilizzato per assumere un ruolo in un altro account con il concatenamento dei ruoli.

Come amministratore dell'account, immagina di voler consentire all'utente IAM `DevUser` nel tuo account per assumere il `Developer_Role` nello stesso account. Tuttavia, si desidera consentire questa operazione solo se l'utente ha impostato l'identità di origine sul proprio nome utente IAM. La seguente policy può essere collegata a un utente IAM.

Example Esempio di politica basata sull'identità allegata a DevUser

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/Developer_Role"
    },
    {
      "Sid": "SetAwsUserNameAsSourceIdentity",
      "Effect": "Allow",
      "Action": "sts:SetSourceIdentity",
      "Resource": "arn:aws:iam::123456789012:role/Developer_Role",
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": "${aws:username}"
        }
```

```
      }
    }
  ]
}
```

Per applicare i valori di identità di origine accettabili, è possibile configurare la policy di attendibilità del ruolo riportata di seguito. La policy fornisce all'utente IAM le autorizzazioni `DevUser`per assumere il ruolo e impostare un'identità di origine. La chiave di condizione `sts:SourceIdentity` definisce il valore di identità di origine accettabile.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDevUserAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/DevUser"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "sts:SourceIdentity": "DevUser"
        }
      }
    }
  ]
}
```

Utilizzando le credenziali per l'utente IAM`DevUser`, l'utente tenta di presumere l'`DeveloperRole`utilizzo della seguente richiesta. AWS CLI

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Developer_Role \
--role-session-name Dev-project \
```

```
--source-identity DevUser \
```

Quando AWS valuta la richiesta, il contesto della richiesta contiene il `sts:SourceIdentity` comando di. `DevUser`

Specifica di un'identità di origine quando si assume un ruolo

È possibile specificare un'identità di origine quando si utilizza una delle operazioni AWS STS `AssumeRole*` API per ottenere credenziali di sicurezza temporanee per un ruolo. L'operazione API che usi è diversa a seconda del caso d'uso. Ad esempio, se utilizzi i ruoli IAM per consentire agli utenti IAM di accedere a AWS risorse a cui normalmente non hanno accesso, potresti utilizzare l'`AssumeRole`operazione. Se utilizzi la federazione delle identità aziendali per gestire gli utenti della forza lavoro, puoi utilizzare l'operazione `AssumeRoleWithSAML`. Se utilizzi la federazione OIDC per consentire agli utenti finali di accedere alle tue applicazioni mobili o web, potresti utilizzare l'`AssumeRoleWithWebIdentity`operazione. Nelle sezioni seguenti viene illustrato come utilizzare l'identità di origine per ogni operazione. Per ulteriori informazioni sugli scenari comuni per le credenziali temporanee, consulta Scenari comuni per le credenziali temporanee.

Utilizzo dell'identità di origine con AssumeRole

L'`AssumeRole`operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. È possibile utilizzare le credenziali dell'utente o del ruolo IAM per chiamare `AssumeRole`. Per passare l'identità di origine mentre assumi un ruolo, utilizzate l'`--source-identity` AWS CLI opzione o il parametro `SourceIdentity` AWS API. L'esempio seguente illustra come specificare l'identità di origine utilizzando la AWS CLI.

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/developer \
--role-session-name Audit \
--source-identity Admin \
```

Utilizzo dell'identità di origine con SAML AssumeRoleWith

Il principale che richiama l'operazione `AssumeRoleWithSAML` viene autenticato utilizzando la federazione basata su SAML. Questa operazione restituisce un set di credenziali temporanee che puoi utilizzare per accedere AWS alle risorse. Per ulteriori informazioni sull'utilizzo della federazione

basata su SAML per AWS Management Console l'accesso, consulta. Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console Per dettagli sull'accesso alle AWS CLI nostre AWS API, consulta. Federazione SAML 2.0 Per un tutorial sulla configurazione della federazione SAML per gli utenti di Active Directory, consulta AWS Federated Authentication with Active Directory Federation Services (ADFS) nel AWS Security Blog.

In qualità di amministratore, puoi consentire ai membri della directory aziendale di unirsi AWS per utilizzare l'operazione. AWS STS `AssumeRoleWithSAML` A tale scopo, è necessario completare le seguenti attività:

1. Configura un provider SAML nella tua organizzazione.
2. Creazione di un provider SAML in IAM.
3. Configura un ruolo e le relative autorizzazioni AWS per i tuoi utenti federati.
4. Fine della configurazione del provider di identità SAML e creazione di asserzioni per la risposta di autenticazione SAML

Per impostare un attributo SAML per l'identità di origine, includi l'elemento `Attribute` con l'attributo `Name` impostato su `https://aws.amazon.com/SAML/Attributes/SourceIdentity`. Utilizza l'elemento `AttributeValue` per specificare il valore dell'identità di origine. Ad esempio, si supponga di voler passare i seguenti attributi di identità come identità di origine:

```
SourceIdentity:DiegoRamirez
```

Per passare questi attributi, includi i seguenti elementi nell'asserzione SAML.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SourceIdentity">
<AttributeValue>DiegoRamirez</AttributeValue>
</Attribute>
```

Utilizzo dell'identità di origine con AssumeRoleWithWebIdentity

La chiamata principale all'`AssumeRoleWithWebIdentity`operazione viene autenticata utilizzando la federazione conforme a OpenID Connect (OIDC). Questa operazione restituisce un insieme di credenziali temporanee che è possibile utilizzare per accedere alle risorse di AWS . Per ulteriori informazioni sull'utilizzo della federazione OIDC per l'accesso, vedere. AWS Management Console Federazione OIDC

Per passare l'identità di origine da OpenID Connect (OIDC), è necessario includere l'identità di origine nel Token Web JSON (JWT). Includi l'identità di origine nello spazio dei nomi `https://aws.amazon.com/ source_identity` nel token quando si invia la richiesta `AssumeRoleWithWebIdentity`. Per ulteriori informazioni sui token e le registrazioni OIDC, consultare Utilizzo di token con pool di utenti nella Guida per gli sviluppatori Amazon Cognito .

Ad esempio, il seguente JWT decodificato è un token utilizzato per chiamare `AssumeRoleWithWebIdentity` con l'identità di origine `Admin`.

Example Esempio di token Web JSON decodificato

```
{
    "sub": "johndoe",
    "aud": "ac_oic_client",
    "jti": "ZYUCeRMQVtqHypVPWAN3VB",
    "iss": "https://xyz.com",
    "iat": 1566583294,
    "exp": 1566583354,
    "auth_time": 1566583292,
    "https://aws.amazon.com/source_identity":"Admin"
}
```

Controllo dell'accesso tramite le informazioni sull'identità di origine

Quando viene inizialmente impostata un'identità di origine, la SourceIdentity chiave sts: è presente nella richiesta. Dopo aver impostato un'identità di origine, la SourceIdentity chiave aws: è presente in tutte le richieste successive effettuate durante la sessione di ruolo. In qualità di amministratore, puoi scrivere politiche che concedano l'autorizzazione condizionale per eseguire AWS azioni in base all'esistenza o al valore dell'attributo di identità di origine.

Immagina di voler richiedere ai tuoi sviluppatori di impostare un'identità di origine per assumere un ruolo fondamentale con il permesso di scrivere su una AWS risorsa fondamentale per la produzione. Immaginate inoltre di concedere AWS l'accesso alle identità della vostra forza lavoro utilizzando. `AssumeRoleWithSAML` Desideri solo che gli sviluppatori senior Saanvi e Diego abbiano accesso al ruolo, in modo che possano creare le seguenti policy di attendibilità per il ruolo.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine (SAML)

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "SAMLProviderAssumeRoleWithSAML",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:saml-provider/name-of-identity-
provider"
      },
      "Action": [
        "sts:AssumeRoleWithSAML"
      ],
      "Condition": {
        "StringEquals": {
          "SAML:aud": "https://signin.aws.amazon.com/saml"
        }
      }
    },
    {
      "Sid": "SetSourceIdentitySrEngs",
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:saml-provider/name-of-identity-
provider"
      },
      "Action": [
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "sts:SourceIdentity": [
            "Saanvi",
            "Diego"
          ]
        }
      }
    }
  ]
}
```

La policy di attendibilità contiene una condizione per `sts:SourceIdentity` che richiede un'identità di origine impostata su Saanvi o Diego per assumere il ruolo critico.

In alternativa, se utilizzi un provider OIDC per la federazione e gli utenti sono autenticati con essoAssumeRoleWithWebIdentity, la tua politica di fiducia dei ruoli potrebbe essere la seguente.

Example Esempio di policy di attendibilità dei ruoli dell'identità di origine (provider OIDC)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/server.example.com"
      },
      "Action": [
        "sts:AssumeRoleWithWebIdentity",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "server.example.com:aud": "oidc-audience-id"
        },
        "StringLike": {
          "sts:SourceIdentity": [
            "Saanvi",
            "Diego"
          ]
        }
      }
    }
  ]
}
```

Concatenamento dei ruoli e requisiti tra account

Immagina di voler consentire agli utenti che hanno assunto il ruolo `CriticalRole` di assumere un ruolo `CriticalRole_2` in un altro account. Le credenziali della sessione del ruolo ottenute per assumere `CriticalRole` sono utilizzate per il concatenamento dei ruoli per un secondo ruolo, `CriticalRole_2`, in un account diverso. Il ruolo viene assunto oltre un limite di account. Pertanto, l'autorizzazione `sts:SetSourceIdentity` deve essere concessa in entrambe le policy di autorizzazione su `CriticalRole` e nella policy di attendibilità del ruolo su `CriticalRole_2`.

Example Esempio di politica di autorizzazione su CriticalRole

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "AssumeRoleAndSetSourceIdentity",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Resource": "arn:aws:iam::222222222222:role/CriticalRole_2"
    }
  ]
}
```

Per proteggere l'impostazione dell'identità di origine attraverso il limite dell'account, la seguente policy di attendibilità del ruolo considera attendibile solo il principale del ruolo per `CriticalRole` per impostare l'identità di origine.

Example Esempio di politica di fiducia dei ruoli su _2 CriticalRole

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:role/CriticalRole"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": ["Saanvi","Diego"]
        }
      }
    }
  ]
}
```

L'utente effettua la chiamata seguente utilizzando le credenziali della sessione di ruolo ottenute da assumendo. CriticalRole L'identità di origine è stata impostata durante l'assunzione di CriticalRole,

quindi non è necessario impostarla nuovamente in modo esplicito. Se l'utente prova a impostare un'identità di origine diversa dal set di valori quando è stato assunto `CriticalRole`, la richiesta di assumere il ruolo verrà rifiutata.

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::222222222222:role/CriticalRole_2 \
--role-session-name Audit \
```

Quando il principale chiamante assume il ruolo, l'identità di origine nella richiesta persiste dalla prima sessione del ruolo assunto. Pertanto, entrambe le chiavi `aws:SourceIdentity` e `sts:SourceIdentity` sono presenti nel contesto della richiesta.

Visualizzazione dell'identità di origine in CloudTrail

È possibile utilizzare CloudTrail per visualizzare le richieste fatte per assumere ruoli o federare gli utenti. È inoltre possibile visualizzare le richieste di ruoli o utenti per eseguire operazioni in AWS. Il file di CloudTrail registro include informazioni sull'identità di origine impostata per il ruolo assunto o la sessione utente federata. Per ulteriori informazioni, consulta Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail

Ad esempio, supponiamo che un utente effettui una AWS STS `AssumeRole` richiesta e imposti un'identità di origine. Puoi trovare le `sourceIdentity` informazioni nella `requestParameters` chiave del tuo CloudTrail registro.

Example Esempio di sezione RequestParameters in un registro AWS CloudTrail

```
"eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AIDAJ45Q7YFFAREXAMPLE",
        "accountId": "111122223333"
    },
    "eventTime": "2020-04-02T18:20:53Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "203.0.113.64",
    "userAgent": "aws-cli/1.16.96 Python/3.6.0 Windows/10 botocore/1.12.86",
    "requestParameters": {
        "roleArn": "arn:aws:iam::123456789012:role/DevRole",
        "roleSessionName": "Dev1",
        "sourceIdentity": "source-identity-value-set"
    }
```

Se l'utente utilizza la sessione di ruolo presunta per eseguire un'azione, le informazioni sull'identità di origine sono presenti nella `userIdentity` chiave del CloudTrail registro.

Example Esempio di chiave UserIdentity in un registro AWS CloudTrail

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAJ45Q7YFFAREXAMPLE:Dev1",
    "arn": "arn:aws:sts::123456789012:assumed-role/DevRole/Dev1",
    "accountId": "123456789012",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAJ45Q7YFFAREXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/DevRole",
        "accountId": "123456789012",
        "userName": "DevRole"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-02-21T23:46:28Z"
      },
      "sourceIdentity": "source-identity-value-present"
    }
  }
}
```

Per vedere esempi di eventi AWS STS API nei CloudTrail log, consulta. Esempi di eventi dell'API IAM nel CloudTrail registro Per maggiori dettagli sulle informazioni contenute nei file di CloudTrail registro, consulta CloudTrail Event Reference nella Guida per l'AWS CloudTrail utente.

Autorizzazioni per GetFederationToken

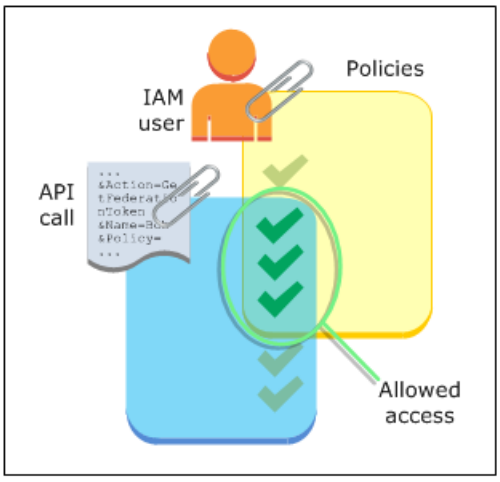
L'operazione `GetFederationToken` viene chiamata da un utente IAM e restituisce le credenziali temporanee per tale utente. Questa operazione consolida l'utente. Le autorizzazioni assegnate a un utente federato sono definite in una di due posizioni:

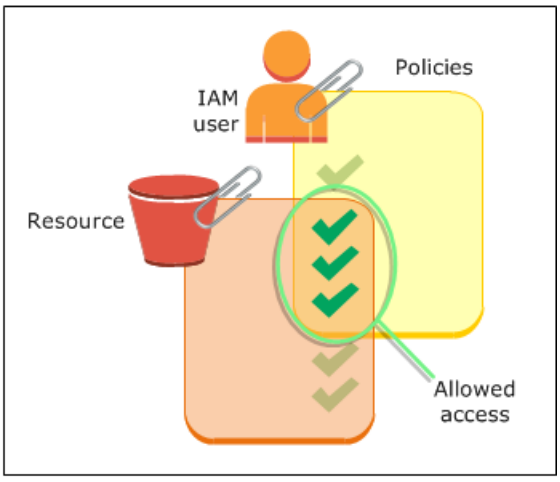
- Le policy di sessione passate come un parametro della chiamata API `GetFederationToken`. (Questo è più comune).
- Una policy basate sulle risorse che nomina esplicitamente l'utente federato nell'elemento `Principal` della policy. (Questo è meno comune).

Le policy di sessione sono policy avanzate che vengono passate come parametri quando si crea una sessione temporanea a livello di programma. Quando crei una sessione per l'utente federato e passi le policy di sessione, le autorizzazioni della sessione risultante sono l'intersezione della policy basata su identità dell'utente e le policy di sessione. Non puoi utilizzare la policy di sessione per concedere autorizzazioni maggiori rispetto a quelle consentite dalla policy basata su identità dell'utente che viene federato.

Nella maggior parte dei casi, se non si passa una policy con la chiamata API `GetFederationToken`, le credenziali di sicurezza temporanee risultanti non dispongono di autorizzazioni. Tuttavia, una policy basata sulle risorse è in grado di fornire ulteriori autorizzazioni per la sessione. Puoi accedere a una risorsa con una policy basata sulle risorse che specifica la sessione come l'entità principale consentita.

Le seguenti immagini mostrano una rappresentazione visiva di come le policy interagiscono per determinare le autorizzazioni per le credenziali di sicurezza provvisorie restituite da una chiamata a `GetFederationToken`.





Esempio: assegnazione di autorizzazioni utilizzando GetFederationToken

È possibile utilizzare l'operazione API `GetFederationToken` con diversi tipi di policy. Di seguito sono illustrati alcuni esempi.

Policy collegata all'utente IAM

In questo esempio, disponi di un'applicazione client basata sul browser che si avvale di due servizi Web di back-end. Un servizio di back-end è il tuo server di autenticazione che utilizza un sistema di identità per autenticare l'applicazione client. L'altro servizio di back-end è un servizio AWS che fornisce alcune delle funzionalità dell'applicazione client. L'applicazione client viene autenticata mediante il tuo server, il quale crea o recupera la policy di autorizzazione appropriata. Il server chiama l'API `GetFederationToken` per ottenere le credenziali di sicurezza provvisorie e restituisce tali credenziali all'applicazione client. L'applicazione client può quindi effettuare richieste direttamente al AWS servizio con le credenziali di sicurezza temporanee. Questa architettura consente all'applicazione client di effettuare AWS richieste senza incorporare credenziali a lungo termine AWS .

Il tuo server di autenticazione chiama l'API `GetFederationToken` con le credenziali di sicurezza a lungo termine di un utente IAM denominato `token-app`. Tuttavia, le credenziali utente IAM a lungo termine rimangono nel server e non vengono mai distribuite al client. La seguente policy di esempio è collegata all'utente `token-app` IAM e definisce la più ampia gamma di autorizzazioni di cui gli utenti federati (client) avranno bisogno. Si noti che l'autorizzazione `sts:GetFederationToken` è necessaria per il servizio di autenticazione per ottenere le credenziali di sicurezza provvisorie per gli utenti federati.

> **Note**
>
> AWS fornisce un'applicazione Java di esempio per questo scopo, che è possibile scaricare qui: Token Vending Machine for Identity Registration - Sample Java Web Application.

Example Esempio di policy collegata all'utente IAM **`token-app`** che chiama **`GetFederationToken`**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": "sts:GetFederationToken",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "dynamodb:ListTables",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "sqs:ReceiveMessage",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "sns:ListSubscriptions",
      "Resource": "*"
    }
  ]
}
```

La policy precedente concede diverse autorizzazioni all'utente IAM. Tuttavia, questa policy da sola non concede alcuna autorizzazione all'utente federato. Se questo utente IAM chiama `GetFederationToken` e non passa una policy come un parametro della chiamata API, l'utente federato risultante non disporrà di autorizzazioni valide.

Policy di sessione passata come parametro

Il modo più comune per assicurare che all'utente federato vengano assegnate le autorizzazioni appropriate è quello di passare una policy di sessione nella chiamata API `GetFederationToken`. Sulla base dell'esempio precedente, immagina che `GetFederationToken` venga chiamato con le credenziali dell'utente IAM `token-app`. Quindi, immagina che la policy di sessione seguente venga passata come un parametro della chiamata API. L'utente federato risultante dispone dell'autorizzazione per elencare i contenuti del bucket Amazon S3 denominato `productionapp`. L'utente non può eseguire le operazioni `GetObject`, `PutObject` e `DeleteObject` di Amazon S3 su elementi nel bucket `productionapp`.

All'utente federato vengono assegnate queste autorizzazioni perché le autorizzazioni sono l'intersezione delle policy utente IAM e delle policy di sessione che vengono passate.

L'utente federato potrebbe non eseguire operazioni in Amazon SNS, Amazon SQS, Amazon DynamoDB o qualsiasi bucket S3 tranne `productionapp`. Queste operazioni sono rifiutate anche se tali autorizzazioni sono concesse all'utente IAM associato alla chiamata `GetFederationToken`.

Example Esempio di policy di sessione passata come parametro della chiamata API **`GetFederationToken`**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::productionapp"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::productionapp/*"]
    }
  ]
}
```

Policy basate su risorse

Alcune AWS risorse supportano politiche basate sulle risorse e queste politiche forniscono un altro meccanismo per concedere le autorizzazioni direttamente a un utente federato. Solo alcuni servizi supportano politiche basate sulle risorse AWS . Ad esempio, Amazon S3 ha i bucket, Amazon SNS ha gli argomenti e Amazon SQS ha le code, tutti elementi ai quali è possibile collegare le policy. Per un elenco di tutti i servizi che supportano le policy basate su risorse, consulta AWS servizi che funzionano con IAM e analizza la colonna "Policy basate su risorse" delle tabelle. Puoi usare policy basate sulle risorse per assegnare le autorizzazioni direttamente a un utente federato. A questo scopo, specifica l'Amazon Resource Name (ARN) dell'utente federato nell'elemento `Principal`

della policy basata sulle risorse. Ciò viene illustrato nell'esempio seguente espandendo gli esempi precedenti e utilizzando un bucket S3 denominato `productionapp`.

La policy basata sulle risorse riportata di seguito è collegata al bucket. La policy di questo bucket consente a un utente federato di nome Carol di accedere al bucket. Quando la policy di esempio descritta in precedenza è collegata all'utente `token-app` IAM, l'utente federato di nome Carol dispone dell'autorizzazione per eseguire le operazioni `s3:GetObject`, `s3:PutObject` e `s3:DeleteObject` sul bucket denominato `productionapp`. Questo vale anche quando nessuna policy di sessione viene passata come parametro della chiamata API `GetFederationToken`. Questo perché in questo caso l'utente federato che si chiama Carol ha ottenuto esplicitamente le autorizzazioni dalla seguente policy basate su risorse.

Ricorda che a un utente federato vengono concesse le autorizzazioni solo quando tali autorizzazioni vengono concesse esplicitamente sia all'utente IAM che all'utente federato. Possono essere concesse (all'interno dell'account) anche da una policy basata su risorse che nomini esplicitamente l'utente federato nell'elemento `Principal` della policy, come nell'esempio seguente.

Example Esempio di policy del bucket che consente l'accesso all'utente federato

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": {"AWS": "arn:aws:sts::account-id:federated-user/Carol"},
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
    ],
    "Resource": ["arn:aws:s3:::productionapp/*"]
  }
}
```

Per ulteriori informazioni su come vengono valutate le policy, consulta la sezione Logica di valutazione delle policy.

Autorizzazioni per GetSessionToken

La principale occasione per chiamare l'operazione API `GetSessionToken` o il comando CLI `get-session-token` è quando un utente deve essere autenticato tramite Multi-Factor Authentication

(MFA). È possibile scrivere una policy che permette determinate operazioni solo quando tali operazioni vengono richieste da un utente autenticato con MFA. Per superare i controlli di autorizzazione MFA, un utente deve prima chiamare `GetSessionToken` e includere i parametri facoltativi `SerialNumber` e `TokenCode`. Se l'utente è stato autenticato con un dispositivo MFA, le credenziali restituite dall'operazione API `GetSessionToken` includono il contesto MFA. Tale contesto indica che l'utente viene autenticato con MFA ed è autorizzato per le operazioni API che richiedono l'autenticazione MFA.

Autorizzazioni richieste per GetSessionToken

Non è richiesta all'utente alcuna autorizzazione per ottenere un token di sessione. Lo scopo dell'operazione `GetSessionToken` è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare le operazioni di autenticazione.

Per concedere le autorizzazioni necessarie per eseguire la maggior parte AWS delle operazioni, è necessario aggiungere l'azione con lo stesso nome a una policy. Ad esempio, per creare un utente, occorre utilizzare l'operazione API `CreateUser`, il comando della CLI `create-user` o la AWS Management Console. Per eseguire queste operazioni, è necessario disporre di una policy che consenta di accedere all'operazione `CreateUser`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateUser",
            "Resource": "*"
        }
    ]
}
```

È possibile includere l'operazione `GetSessionToken` nella policy, ma questo non incide sulla capacità di un utente di eseguire l'operazione `GetSessionToken`.

Autorizzazioni concesse da GetSessionToken

Se la chiamata a `GetSessionToken` viene eseguita con le credenziali di un utente IAM, le credenziali di sicurezza temporanee avranno le stesse autorizzazioni dell'utente IAM. Analogamente, se `GetSessionToken` viene chiamata con Utente root dell'account AWS credenziali, le credenziali di sicurezza temporanee dispongono dei permessi di utente root.

> **Note**
>
> È consigliabile non chiamare `GetSessionToken` con credenziali dell'utente root Segui invece le best practice e crea utenti IAM con le autorizzazioni necessarie. Quindi usa questi utenti IAM per l'interazione quotidiana con. AWS

Le credenziali temporanee ottenute chiamando `GetSessionToken` hanno le funzionalità e le limitazioni seguenti:

- Puoi utilizzare le credenziali per accedere a passando le credenziali all' AWS Management Console endpoint Single Sign-On della federazione all'indirizzo. `https://signin.aws.amazon.com/federation` Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.
- Non puoi utilizzare le credenziali per richiamare le operazioni API IAM o AWS STS . È possibile utilizzarle per chiamare le operazioni API per altri servizi. AWS

Per un confronto tra questa operazione API e i relativi limiti e funzionalità con le altre operazioni API che creano credenziali di sicurezza temporanee, consulta Confronto delle operazioni AWS STS API

Per ulteriori informazioni sull'accesso API protetto da MFA con `GetSessionToken`, consulta Configurazione dell'accesso alle API protetto da MFA.

Disabilitazione delle autorizzazioni per le credenziali di sicurezza temporanee

Le credenziali di sicurezza provvisorie sono valide finché non scadono. Queste credenziali sono valide per la durata specificata, da 900 secondi (15 minuti) a un massimo di 129.600 secondi (36 ore). La durata predefinita della sessione è di 43.200 secondi (12 ore). Puoi revocare queste credenziali, ma devi anche modificare le autorizzazioni per il ruolo per bloccare l'uso di credenziali compromesse che consentirebbero attività dannose per l'account. Le autorizzazioni assegnate alle credenziali di sicurezza temporanee vengono valutate ogni volta che vengono utilizzate per effettuare una richiesta. AWS Una volta rimosse tutte le autorizzazioni dalle credenziali, le AWS richieste che le utilizzano hanno esito negativo.

Potrebbero essere necessari alcuni minuti prima che gli aggiornamenti delle politicy siano applicati. Revoca le credenziali di sicurezza temporanee del ruolo per obbligare tutti gli utenti che assumono il ruolo a riautenticarsi e richiedere nuove credenziali.

Non è possibile modificare le autorizzazioni per un. Utente root dell'account AWS Analogamente, non puoi modificare le autorizzazioni relative alle credenziali di sicurezza temporanee create richiamando `GetFederationToken` o `GetSessionToken` mentre sei collegato come utente root. Per questo motivo, consigliamo di non effettuare la chiamata a `GetFederationToken` o `GetSessionToken` come utente root.

> **Important**
>
> Non è possibile modificare i ruoli in IAM creati dai set di autorizzazioni di IAM Identity Center. È necessario revocare la sessione attiva del set di autorizzazioni per un utente in IAM Identity Center. Per ulteriori informazioni, consulta Revoca le sessioni di ruolo IAM attive create dai set di autorizzazioni nella Guida per l'utente di IAM Identity Center.

Argomenti

- Negare l'accesso a tutte le sessioni associate a un ruolo
- Negare l'accesso a una sessione specifica
- Negare una sessione utente con le chiavi di contesto delle condizioni
- Rifiutare un utente di sessione con policy basate sulle risorse

Negare l'accesso a tutte le sessioni associate a un ruolo

Usa questo approccio quando hai il dubbio che sia stato effettuato un accesso sospetto da:

- Principali di un altro account utilizzando l'accesso multi-account
- Identità utente esterne con autorizzazioni per accedere AWS alle risorse del tuo account
- Utenti che sono stati autenticati in un'applicazione mobile o web con un provider OIDC

Questa procedura nega le autorizzazioni a tutti gli utenti che dispongono delle autorizzazioni per assumere un ruolo.

Per modificare o rimuovere le autorizzazioni assegnate alle credenziali di sicurezza provvisorie ottenute richiamando `AssumeRole`, `AssumeRoleWithSAML`, o `AssumeRoleWithWebIdentity`, `GetFederationToken` o `GetSessionToken`, puoi modificare o eliminare la policy di autorizzazione che definisce le autorizzazioni per il ruolo.

> **Important**
>
> Se esiste una policy basata sulle risorse che consente l'accesso principale, devi anche aggiungere un rifiuto esplicito per quella risorsa. Per informazioni dettagliate, vedi Rifiutare un utente di sessione con policy basate sulle risorse.

1. Accedi AWS Management Console e apri la console IAM.
2. Nel riquadro di navigazione, seleziona il nome del ruolo da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco.
3. Seleziona la policy pertinente.
4. Scegli la scheda Autorizzazioni.
5. Scegli la scheda JSON e aggiorna la policy per negare tutte le risorse e le azioni.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

6. Nella pagina Review (Esamina) controllare Summary (Riepilogo) e selezionare Save changes (Salva modifiche) per salvare.

Quando si modifica o si elimina la policy, le modifiche riguardano le autorizzazioni di tutte le credenziali di sicurezza provvisorie associate a tale ruolo, tra cui credenziali che sono state rilasciate prima di aver modificato la policy di autorizzazione del ruolo. Dopo aver aggiornato la policy, è possibile revocare le credenziali di sicurezza temporanee del ruolo per revocare immediatamente tutte le autorizzazioni alle credenziali emesse per il ruolo.

Negare l'accesso a una sessione specifica

Quando aggiorni i ruoli assumibili da un IdP con una policy di negazione totale o elimini completamente il ruolo, tutti gli utenti che hanno accesso al ruolo vengono scollegati Puoi negare

l'accesso in base all'elemento `Principal` senza influire sulle autorizzazioni di tutte le altre sessioni associate al ruolo.

Ai `Principal` possono essere negate le autorizzazioni usando chiavi di contesto delle condizioni o policy basate sulle risorse.

> **Tip**
>
> Puoi trovare gli ARN degli utenti federati utilizzando AWS CloudTrail i log. Per ulteriori informazioni, consulta Come identificare facilmente gli utenti federati utilizzando. AWS CloudTrail

Negare una sessione utente con le chiavi di contesto delle condizioni

Puoi usare chiavi del contesto della condizione in situazioni in cui vuoi negare l'accesso a sessioni specifiche con credenziali di sicurezza temporanee senza compromettere le autorizzazioni dell'utente o dell'utente IAM che ha creato le credenziali.

Per ulteriori informazioni su queste chiavi di contesto della condizione, consulta AWS chiavi di contesto della condizione globale.

> **Note**
>
> Se esiste una policy basata sulle risorse che consente l'accesso principale, devi anche aggiungere una istruzione di negazione esplicita sulla policy basata sulle risorse dopo aver completato questi passaggi.

Dopo aver aggiornato la policy, puoi revocare le credenziali di sicurezza temporanee del ruolo per revocare immediatamente tutte le credenziali emesse.

leggi: PrincipalArn

Puoi usare la chiave di contesto della condizione Leggi: PrincipalArn per negare l'accesso a uno specifico ARN principale. A tale scopo, devi specificare l'identificatore univoco (ID) dell'utente, del ruolo o dell'utente federato IAM a cui sono associate le credenziali di sicurezza temporanee nell'elemento condizione di una policy.

1. Nel pannello di navigazione della console IAM seleziona il nome del ruolo da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco.
2. Seleziona la policy pertinente.
3. Scegli la scheda Autorizzazioni.
4. Scegli la scheda JSON e aggiungi un'istruzione di negazione per l'ARN principale, come mostrato nell'esempio seguente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:role/ROLENAME",
            "arn:aws:iam::222222222222:user/USERNAME",
            "arn:aws:sts::222222222222:federated-user/USERNAME"
          ]
        }
      }
    }
  ]
}
```

5. Nella pagina Review (Esamina) controllare Summary (Riepilogo) e selezionare Save changes (Salva modifiche) per salvare.

aws:userid

Puoi usare la chiave di contesto della condizione aws:userid per negare l'accesso a tutte o a specifiche sessioni con credenziali di sicurezza temporanee associate all'utente o al ruolo IAM. A tale scopo, devi specificare l'identificatore univoco (ID) dell'utente IAM, del ruolo o dell'utente federato IAM a cui sono associate le credenziali di sicurezza temporanee nell'elemento `Condition` di una policy.

La seguente policy mostra un esempio di come puoi negare l'accesso a sessioni con credenziali di sicurezza temporanee utilizzando la chiave di contesto della condizione.`aws:userid`.

- `AIDAXUSER1` rappresenta l'identificatore univoco per un utente IAM. Specificando l'identificatore univoco di un utente IAM come valore per la chiave di contesto `aws:userid` verranno negate tutte le sessioni associate all'utente IAM.
- `AROAXROLE1` rappresenta l'identificatore univoco per un ruolo IAM. Specificando l'identificatore univoco di un ruolo IAM come valore per la chiave di contesto `aws:userid` verranno negate tutte le sessioni associate al ruolo IAM.
- `AROAXROLE2` rappresenta l'identificatore univoco per una sessione del ruolo assunto. Nella parte caller-specified-role-session -name dell'identificatore univoco del ruolo assunto è possibile specificare un nome di sessione di ruolo o un carattere jolly se viene utilizzato l'operatore condition. StringLike Se specifichi il nome di sessione del ruolo, verrà negata la sessione di ruolo denominata senza influire sulle autorizzazioni del ruolo che ha creato le credenziali. Se specifichi un carattere jolly per il nome di sessione del ruolo, verranno negate tutte le sessioni associate al ruolo.
- `account-id:<federated-user-caller-specified-name>` rappresenta l'identificatore univoco per una sessione di un utente federato. Un utente federato viene creato da un utente IAM che chiama l'API. GetFederationToken Se specifichi l'identificatore univoco per un utente federato, verrà negata la sessione dell'utente federato denominata senza influire sulle autorizzazioni del ruolo che ha creato le credenziali.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:userId": [
            "AIDAXUSER1",
            "AROAXROLE1",
            "AROAXROLE2:<caller-specified-role-session-name>",
            "account-id:<federated-user-caller-specified-name>"
          ]
        }
      }
    }
  ]
}
```

Per esempi specifici di valori chiave del principale, consulta Valori della chiave dell'entità principale. Per ulteriori informazioni sugli identificatori univoci IAM, consulta Identificatori univoci.

Rifiutare un utente di sessione con policy basate sulle risorse

Se l'ARN principale è incluso anche in una policy basata sulle risorse, devi revocare l'accesso sulla base dei valori `principalId` o `sourceIdentity` dell'utente specifico nell'elemento `Principal` di una policy basata sulle risorse. Se aggiorni solo la policy delle autorizzazioni per il ruolo, l'utente può comunque eseguire le azioni consentite nella policy basata sulle risorse.

1. Fai riferimento a AWS servizi che funzionano con IAM per verificare se il servizio supporta policy basate sulle risorse.
2. Accedi AWS Management Console e apri la console per il servizio. Ogni servizio ha una posizione diversa nella console per allegare le policy.
3. Modifica l'informativa sulla policy per specificare le informazioni identificative delle credenziali:
   a. In `Principal`, inserisci l'ARN delle credenziali da negare.
   b. In `Effect`, inserisci "Nega".
   c. In `Action`, inserisci lo spazio dei nomi del servizio e il nome dell'azione da rifiutare. Per negare tutte le azioni, usa il carattere jolly (*). Ad esempio: "s3:*".
   d. In `Resource`, inserisci l'ARN della risorsa di destinazione. Ad esempio: "arn:aws:s3:::EXAMPLE-BUCKET".

```
{
"Version": "2012-10-17",
  "Statement": {
    "Principal": [
            "arn:aws:iam::222222222222:role/ROLENAME",
            "arn:aws:iam::222222222222:user/USERNAME",
            "arn:aws:sts::222222222222:federated-user/USERNAME"
    ],
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::EXAMPLE-BUCKET"
  }
}
```

4. Salva il tuo lavoro.

Concessione delle autorizzazioni per creare credenziali di sicurezza temporanee

Per impostazione predefinita, gli utenti IAM non dispongono dell'autorizzazione per creare credenziali di sicurezza temporanee per ruoli e utenti federati. È necessario utilizzare una policy per fornire queste autorizzazioni agli utenti. Anche se è possibile concedere le autorizzazioni direttamente a un utente, ti consigliamo caldamente di assegnarle a un gruppo. In questo modo la gestione delle autorizzazioni risulta molto più semplice. Quando un utente non ha più bisogno di eseguire le operazioni associate alle autorizzazioni, non dovrai fare altro che rimuoverlo dal gruppo. Se un'altra persona si trova nella necessità di eseguire tale operazione, sarà sufficiente aggiungerla al gruppo per concederle le autorizzazioni.

Per concedere a un gruppo IAM l'autorizzazione per creare credenziali di sicurezza temporanee per ruoli o utenti federati, puoi collegare una policy che conceda uno o entrambi i seguenti privilegi:

- Per consentire agli utenti federati di accedere a un ruolo IAM, concedi l'accesso a AWS STS `AssumeRole`.
- Per gli utenti federati che non necessitano di un ruolo, concedi l'accesso a. AWS STS `GetFederationToken`

Per informazioni sulle differenze fra le operazioni API `AssumeRole` e `GetFederationToken`, consultare Richiesta di credenziali di sicurezza temporanee.

Per creare le credenziali di sicurezza temporanee, gli utenti IAM possono chiamare anche `GetSessionToken`. Non sono necessarie autorizzazioni perché un utente possa chiamare `GetSessionToken`. Lo scopo dell'operazione è autenticare l'utente tramite MFA. Non è possibile utilizzare le policy per controllare l'autenticazione. Ciò significa che non puoi impedire agli utenti IAM di chiamare `GetSessionToken` per creare le credenziali temporanee.

Example Esempio di policy che concede autorizzazioni per assumere un ruolo

La politica di esempio seguente concede l'autorizzazione a `AssumeRole` richiedere il `UpdateApp` ruolo in. Account AWS `123123123123` Quando si usa `AssumeRole`, l'utente (o l'applicazione) che crea le credenziali di sicurezza per conto di un utente federato non è in grado di delegare autorizzazioni che non siano già state specificate nella policy di autorizzazione del ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
```

```
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::123123123123:role/UpdateAPP"
  }]
}
```

Example Esempio di policy che concede l'autorizzazione per creare credenziali di sicurezza temporanee per un utente federato.

La policy dell'esempio seguente concede l'autorizzazione per l'accesso a `GetFederationToken`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:GetFederationToken",
    "Resource": "*"
  }]
}
```

Important

Quando si autorizza un utente IAM a creare credenziali di sicurezza temporanee per gli utenti federati con `GetFederationToken`, tale utente avrà la possibilità di delegare le proprie autorizzazioni. Per ulteriori informazioni sulla delega delle autorizzazioni tra utenti IAM e Account AWS, consulta. Esempi di policy per la delega dell'accesso Per informazioni sul controllo delle autorizzazioni nelle credenziali di sicurezza provvisorie, vedi Controllo delle autorizzazioni per le credenziali di sicurezza temporanee.

Example Esempio di policy che concede a un utente autorizzazioni limitate per creare credenziali di sicurezza temporanee per utenti federati.

Quando si consente a un utente IAM di chiamare `GetFederationToken`, una best practice consiste nel limitare le autorizzazioni che l'utente IAM può delegare. Ad esempio, la policy di seguito mostra come consentire a un utente IAM di creare credenziali di sicurezza temporanee solo per gli utenti federati il cui nome inizia con Manager.

```
{
  "Version": "2012-10-17",
  "Statement": [{
```

```
    "Effect": "Allow",
    "Action": "sts:GetFederationToken",
    "Resource": ["arn:aws:sts::123456789012:federated-user/Manager*"]
  }]
}
```

Concessione delle autorizzazioni per l'utilizzo di sessioni di console con riconoscimento dell'identità

Le sessioni di console con riconoscimento dell'identità consentono di includere gli ID AWS IAM Identity Center utente e di sessione nelle sessioni di console degli utenti AWS al momento dell'accesso. Ad esempio, Amazon Q Developer Pro utilizza sessioni di console con riconoscimento dell'identità per personalizzare l'esperienza del servizio. Per ulteriori informazioni sulle sessioni di console con riconoscimento dell'identità, consulta Attivazione delle sessioni di console con riconoscimento dell'identità nella Guida per l'utente.AWS IAM Identity Center Per informazioni sulla configurazione di Amazon Q Developer, consulta Configurazione di Amazon Q Developer nella Amazon Q Developer User Guide.

Affinché le sessioni della console con riconoscimento dell'identità siano disponibili per un utente, è necessario utilizzare una policy basata sull'identità per concedere al responsabile IAM l'`sts:SetContext`autorizzazione per la risorsa che rappresenta la propria sessione di console.

> **Important**
>
> Per impostazione predefinita, gli utenti non sono autorizzati a impostare il contesto per le sessioni di console con riconoscimento dell'identità. Per consentire ciò, è necessario concedere al responsabile IAM l'`sts:SetContext`autorizzazione in una policy basata sull'identità, come illustrato nell'esempio di policy riportato di seguito.

L'esempio seguente di policy basata sull'identità concede l'`sts:SetContext`autorizzazione a un responsabile IAM, che consente al preside di impostare un contesto di sessione della console che riconosca l'identità per le proprie sessioni di console. AWS La risorsa policy,, `arn:aws:sts::`*`account-id`*`:self` rappresenta la sessione del chiamante. AWS Il segmento `account-id` ARN può essere sostituito con un carattere jolly * nei casi in cui la stessa politica di autorizzazione viene implementata su più account, ad esempio quando questa policy viene implementata utilizzando i set di autorizzazioni IAM Identity Center.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:SetContext",
      "Resource": "arn:aws:sts::account-id:self"
    }
  ]
}
```

Gestire AWS STS in un Regione AWS

Per impostazione predefinita, AWS Security Token Service (AWS STS) è disponibile come servizio globale e tutte le AWS STS richieste vanno a un singolo endpoint all'indirizzo. `https://sts.amazonaws.com` AWS consiglia di utilizzare gli AWS STS endpoint regionali anziché l'endpoint globale per ridurre la latenza, aumentare la ridondanza e aumentare la validità dei token di sessione.

- Riduzione della latenza: effettuando AWS STS chiamate verso un endpoint geograficamente più vicino ai servizi e alle applicazioni, è possibile accedere AWS STS a servizi con una latenza inferiore e tempi di risposta migliori.
- Progetta in ridondanza: puoi limitare gli effetti di un guasto all'interno di un carico di lavoro a un numero limitato di componenti con un ambito prevedibile di contenimento degli impatti. L'utilizzo AWS STS degli endpoint regionali consente di allineare l'ambito dei componenti con quello dei token di sessione. Per ulteriori informazioni su questo pilastro di affidabilità, consulta Uso dell'isolamento dei guasti per proteggere il carico di lavoro in Framework AWS Well-Architected.
- Aumenta la validità dei token di sessione: i token di sessione degli AWS STS endpoint regionali sono validi in tutti. Regioni AWS I token di sessione dell'endpoint STS globale sono validi solo se sono abilitati per Regioni AWS impostazione predefinita. Se intendi abilitare una nuova regione per il tuo account, puoi utilizzare i token di sessione dagli endpoint regionali. AWS STS Se scegli di utilizzare l'endpoint globale, devi modificare la compatibilità regionale dei token di AWS STS sessione per l'endpoint globale. In questo modo si garantisce che i token siano validi in tutti. Regioni AWS

Gestione dei token di sessione emessi dall'endpoint globale

Per impostazione predefinita, la Regioni AWS maggior parte è abilitata alle operazioni in Servizi AWS generale. Queste regioni vengono attivate automaticamente per essere utilizzate con AWS STS. Alcune regioni, ad esempio Asia Pacifico (Hong Kong), devono essere abilitate manualmente. Per

ulteriori informazioni sull'attivazione e la disabilitazione Regioni AWS, consulta Specificare le opzioni che Regioni AWS il tuo account può utilizzare nella Guida AWS Account Management di riferimento. Quando abiliti queste AWS regioni, vengono automaticamente attivate per l'uso con AWS STS. Non è possibile attivare l' AWS STS endpoint per una regione disattivata. I token di sessione validi in tutte le aree Regioni AWS includono più caratteri rispetto ai token validi nelle regioni abilitate per impostazione predefinita. La modifica di questa impostazione potrebbe influenzare i sistemi esistenti in cui vengono memorizzati temporaneamente i token.

È possibile modificare questa impostazione utilizzando l'API AWS Management Console, AWS CLI, o AWS .

Modificare le regioni compatibili con i token di sessione l'endpoint globale (console)

1. Accedi come utente root o come utente con le autorizzazioni per eseguire attività di amministrazione di IAM. Per modificare la compatibilità dei token di sessione, è necessario disporre di una policy che consente l'operazione `iam:SetSecurityTokenServicePreferences`.
2. Apri la console IAM. Nel riquadro di navigazione, scegliere Account settings (Impostazioni account).
3. Nella sezione Security Token Service (STS) Token di sessione dagli endpoint STS. L'endpoint globale indica `Valid only in Regioni AWS enabled by default`. Scegliere Change (Cambia).
4. Nella finestra di dialogo Modifica compatibilità dell'area, seleziona Tutto Regioni AWS. Selezionare quindi Save changes (Salva modifiche).

> **Note**
>
> I token di sessione validi in tutte le aree Regione AWS includono più caratteri rispetto ai token validi nelle regioni abilitate per impostazione predefinita. La modifica di questa impostazione potrebbe influenzare i sistemi esistenti in cui vengono memorizzati temporaneamente i token.

Modificare le regioni compatibili con i token di sessione l'endpoint globale (AWS CLI)

Imposta la versione del token di sessione. I token della versione 1 sono validi solo se sono disponibili per impostazione predefinita. Regioni AWS Questi token non funzionano nelle regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le

regioni. Tuttavia, i token versione 2 sono composti da un numero maggiore di caratteri e ciò può influire sui sistemi in cui vengono memorizzati temporaneamente i token.

- `aws iam set-security-token-service-preferences`

Modificare le regioni compatibili con i token di sessione l'endpoint globale (API AWS)

Imposta la versione del token di sessione. I token della versione 1 sono validi solo se sono disponibili per impostazione predefinita. Regioni AWS Questi token non funzionano nelle regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le regioni. Tuttavia, i token versione 2 sono composti da un numero maggiore di caratteri e ciò può influire sui sistemi in cui vengono memorizzati temporaneamente i token.

- `SetSecurityTokenServicePreferences`

AWS STS Attivazione e disattivazione in un Regione AWS

Quando attivi gli endpoint STS per una regione, AWS STS puoi emettere credenziali temporanee agli utenti e ai ruoli del tuo account che effettuano una richiesta. AWS STS Queste credenziali possono essere utilizzate in qualsiasi regione abilitata di default o manualmente. Per le Regioni abilitate per impostazione predefinita, è necessario attivare l'endpoint STS della Regione nell'account in cui vengono generate le credenziali provvisorie. Al momento di effettuare la richiesta, non importa se un utente è autenticato sullo stesso account o su un altro account. Per le Regioni abilitate manualmente, è necessario attivare la Regione sia nell'account che effettua la richiesta sia nell'account in cui vengono generate le credenziali temporanee.

Ad esempio, immagina che un utente dell'account A desideri inviare una richiesta `sts:AssumeRole` API all'endpoint AWS STS regionale. `https://sts.ap-east-1.amazonaws.com` La richiesta è per delle credenziali temporanee per il ruolo denominato `Developer` nell'account B. Poiché la richiesta è di creare le credenziali per un'entità nell'account B, l'account B deve attivare la regione `ap-east-1`. Gli utenti dell'account A (o di qualsiasi altro account) possono chiamare l'endpoint `ap-east-1` AWS STS per richiedere le credenziali per l'account B, che la regione sia attivata o meno nel loro account.

> **Note**
>
> Le regioni attive sono disponibili per tutti gli utenti che utilizzano credenziali provvisorie in tale account. Per controllare quali utenti o ruoli IAM possono accedere alla regione, utilizza la chiave di condizione `aws:RequestedRegion` nelle tue policy di autorizzazione.

Per attivarlo o disattivarlo AWS STS in una regione abilitata per impostazione predefinita (console)

1. Accedi come utente root o come utente con le autorizzazioni per eseguire attività di amministrazione di IAM.
2. Apri la console IAM e, nel pannello di navigazione, seleziona Impostazioni account.
3. Nella sezione Endpoint di Security Token Service (STS), trova la regione che desideri configurare, quindi scegli Active (Attiva) o Inactive (Inattiva) nella colonna STS status (Stato STS).
4. Nella finestra di dialogo visualizzata, scegli Activate (Attiva) o Deactivate (Disattiva).

Per le regioni che devono essere abilitate, ci attiviamo AWS STS automaticamente quando abiliti la regione. Dopo aver abilitato una regione, AWS STS è sempre attiva per la regione e non è possibile disattivarla. Per ulteriori informazioni sull'attivazione delle aree che sono disabilitate per impostazione predefinita, consulta Specificazione delle aree che Regioni AWS il proprio account può utilizzare nella Guida AWS Account Management di riferimento.

Scrittura di codice per l'utilizzo di regioni AWS STS

Dopo aver attivato una regione, puoi indirizzare le chiamate AWS STS API verso quella regione. Il seguente frammento di codice Java mostra come configurare un `AWSSecurityTokenService` oggetto per effettuare richieste alla regione Europa (Milano) (eu-south-1).

```
EndpointConfiguration regionEndpointConfig = new EndpointConfiguration("https://sts.eu-
south-1.amazonaws.com", "eu-south-1");
AWSSecurityTokenService stsRegionalClient =
 AWSSecurityTokenServiceClientBuilder.standard()
.withCredentials(credentials)
.withEndpointConfiguration(regionEndpointConfig)
.build();
```

AWS STS consiglia di effettuare chiamate verso un endpoint regionale. Per informazioni su come abilitare manualmente una regione, consulta Specificare quale regione può essere utilizzata dal Regioni AWS tuo account nella Guida AWS Account Management di riferimento.

Nell'esempio, la prima riga crea un'istanza di un oggetto `EndpointConfiguration` chiamata `regionEndpointConfig`, passando l'URL dell'endpoint e la Regione AWS come parametri.

Per informazioni su come impostare gli endpoint AWS STS regionali utilizzando una variabile di ambiente per gli AWS SDK, consulta Endpoint AWS STS regionalizzati nella Guida di riferimento agli AWS SDK e agli strumenti.

Per tutte le altre combinazioni di linguaggio e ambiente di programmazione, consulta la documentazione dell'SDK pertinente.

Regioni ed endpoint

La seguente tabella elenca le regioni e i relativi endpoint. Indica quali sono attivate per default e quali possono essere attivate o disattivate.

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
--Globale--	sts.amazonaws.com	Sì	No
Stati Uniti orientali (Ohio)	sts.us-east-2.amazonaws.com	Sì	Sì
Stati Uniti orientali (Virginia settentrionale)	sts.us-east-1.amazonaws.com	Sì	No

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Stati Uniti occidentali (California settentrionale)	sts.us-west-1.amazonaws.com	 Sì	 Sì
US West (Oregon)	sts.us-west-2.amazonaws.com	 Sì	 Sì
Africa (Città del Capo)	sts.af-south-1.amazonaws.com	 No[1]	 No
Asia Pacifico (Hong Kong)	sts.ap-east-1.amazonaws.com	 No[1]	 No
Asia Pacific (Hyderabad)	sts.ap-south-2.amazonaws.com	 No[1]	 No
Asia Pacifico (Giacarta)	sts.ap-southeast-3.amazonaws.com	 No[1]	 No

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Asia Pacifico (Melbourne)	sts.ap-southeast-4.amazonaws.com	 No[1]	 No
Asia Pacifico (Mumbai)	sts.ap-south-1.amazonaws.com	 Sì	 Sì
Asia Pacifico (Osaka-Locale)	sts.ap-northeast-3.amazonaws.com	 Sì	 Sì
Asia Pacifico (Seoul)	sts.ap-northeast-2.amazonaws.com	 Sì	 Sì
Asia Pacifico (Singapore)	sts.ap-southeast-1.amazonaws.com	 Sì	 Sì
Asia Pacifico (Sydney)	sts.ap-southeast-2.amazonaws.com	 Sì	 Sì

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Asia Pacifico (Tokyo)	sts.ap-northeast-1.amazonaws.com	 Sì	 Sì
Canada (Centrale)	sts.ca-central-1.amazonaws.com	 Sì	 Sì
Canada occidentale (Calgary)	sts.ca-west-1.amazonaws.com	 Sì	 Sì
Cina (Pechino)	sts.cn-north-1.amazonaws.com.cn	 Sì[1]	 No
Cina (Ningxia)	sts.cn-northwest-1.amazonaws.com.cn	 Sì[1]	 Sì
Europa (Francoforte)	sts.eu-central-1.amazonaws.com	 Sì	 Sì

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Europa (Irlanda)	sts.eu-west-1.amazonaws.com	 Sì	 Sì
Europa (Londra)	sts.eu-west-2.amazonaws.com	 Sì	 Sì
Europa (Milano)	sts.eu-south-1.amazonaws.com	 No[1]	 No
Europa (Parigi)	sts.eu-west-3.amazonaws.com	 Sì	 Sì
Europa (Spagna)	sts.eu-south-2.amazonaws.com	 No[1]	 No
Europa (Stoccolma)	sts.eu-north-1.amazonaws.com	 Sì	 Sì

Nome Regione	Endpoint	Attivo per impostazione predefinita	Attivazione/disattivazione manuale
Europa (Zurigo)	sts.eu-central-2.amazonaws.com	 No[1]	 No
Israele (Tel Aviv)	sts.il-central-1.amazonaws.com	 No[1]	 No
Medio Oriente (Bahrein)	sts.me-south-1.amazonaws.com	 No[1]	 No
Medio Oriente (Emirati Arabi Uniti)	sts.me-central-1.amazonaws.com	 No[1]	 No
Sud America (San Paolo)	sts.sa-east-1.amazonaws.com	 Sì	 Sì

[1]È necessario abilitare la regione per utilizzarla. Ciò attiva automaticamente AWS STS. Non è possibile attivare o disattivare manualmente AWS STS in queste regioni.

[2]Per utilizzarlo AWS in Cina, sono necessari un account e credenziali specifici per la Cina. AWS

AWS CloudTrail e endpoint regionali

Le chiamate agli endpoint regionali e globali vengono registrate sul campo `tlsDetails` in AWS CloudTrail. Le chiamate verso gli endpoint regionali, ad esempio`us-east-2.amazonaws.com`, vengono registrate nella regione CloudTrail appropriata. Le chiamate all'endpoint globale, `sts.amazonaws.com`, vengono registrate come chiamate a un servizio globale. Gli eventi per gli AWS STS endpoint globali vengono registrati su us-east-1.

> **Note**
>
> `tlsDetails` può essere visualizzato solo per i servizi che supportano questo campo. Vedi i dettagli sui servizi che supportano TLS nella Guida per l'utente CloudTrail AWS CloudTrail Per ulteriori informazioni, consulta Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail.

Utilizzo dei token di connessione

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un token AWS STS service bearer prima di poter accedere alle loro risorse a livello di programmazione. Questi servizi supportano un protocollo che richiede l'utilizzo di un token di connessione invece di utilizzare una richiesta firmata Signature Version 4 tradizionale. Quando esegui AWS CLI o esegui operazioni AWS API che richiedono token al portatore, il AWS servizio richiede un token al portatore per tuo conto. Il servizio fornisce il token, che è possibile utilizzare per eseguire le operazioni successive in tale servizio.

AWS STS i service bearer token includono informazioni relative all'autenticazione principale originale che potrebbero influire sulle autorizzazioni dell'utente. Queste informazioni possono includere tag del principale, tag di sessione e policy di sessione. L'ID chiave di accesso del token inizia con il prefisso `ABIA`. Questo ti aiuta a identificare le operazioni che sono state eseguite utilizzando i token service bearer nei tuoi log. CloudTrail

> **Important**
>
> Il token di connessione può essere utilizzato solo per le chiamate al servizio che lo genera e nella regione in cui è stato generato. Non è possibile utilizzare il token di connessione per eseguire operazioni in altri servizi o regioni.

Un esempio di servizio che supporta i token bearer è. AWS CodeArtifact Prima di poter interagire AWS CodeArtifact utilizzando un gestore di pacchetti come NPM, Maven o PIP, è necessario chiamare l'operazione. `aws codeartifact get-authorization-token` Questa operazione restituisce un token portatore che è possibile utilizzare per eseguire operazioni. AWS CodeArtifact In alternativa, è possibile utilizzare il comando `aws codeartifact login` che completa la stessa operazione e quindi configura automaticamente il client.

Se esegui un'azione in un AWS servizio che genera un token bearer per te, devi disporre delle seguenti autorizzazioni nella tua politica IAM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServiceBearerToken",
            "Effect": "Allow",
            "Action": "sts:GetServiceBearerToken",
            "Resource": "*"
        }
    ]
}
```

Per un esempio di token portatore di servizi, consulta Utilizzo di policy basate sulle identità per AWS CodeArtifact nella Guida per l'utente di AWS CodeArtifact.

Applicazioni di esempio che usano credenziali temporanee

È possibile utilizzare AWS Security Token Service (AWS STS) per creare e fornire a utenti attendibili credenziali di sicurezza temporanee in grado di controllare l'accesso alle risorse. AWS Per ulteriori informazioni su AWS STS, consulta Credenziali di sicurezza temporanee in IAM. Per scoprire come è possibile utilizzare AWS STS per gestire le credenziali di sicurezza temporanee, è possibile scaricare le seguenti applicazioni di esempio che implementano scenari di esempio completi:

- Abilitazione della federazione all' AWS utilizzo di Windows Active Directory, ADFS e SAML 2.0. Dimostra come delegare l'accesso utilizzando la federazione aziendale con AWS usando Windows Active Directory (AD), Active Directory Federation Services (ADFS) 2.0 e SAML (Security Assertion Markup Language) 2.0.
- Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console. Dimostra come creare un proxy di federazione personalizzato che abilita l'autenticazione unica

(SSO) in modo che gli utenti esistenti di Active Directory possano accedere alla AWS Management Console.

- Come utilizzare Shibboleth per il Single Sign-On su. AWS Management Console . Mostra come utilizzare Shibboleth e SAML per fornire agli utenti l'accesso Single Sign-On (SSO) alla AWS Management Console.

Esempi per la federazione OIDC

Le seguenti applicazioni di esempio illustrano come utilizzare OIDCFederation con provider come Login with Amazon, Amazon Cognito, Facebook o Google. Puoi scambiare l'autenticazione di questi provider con credenziali di AWS sicurezza temporanee per accedere ai servizi. AWS

- Tutorial Amazon Cognito: ti consigliamo di utilizzare Amazon Cognito con gli SDK per lo sviluppo mobile. AWS Amazon Cognito offre il modo più semplice per gestire l'identità per le applicazioni per dispositivi mobili e offre funzionalità aggiuntive come la sincronizzazione e l'identità tra più dispositivi. Per ulteriori informazioni su Amazon Cognito, consulta Autenticazione con Amplify nella documentazione di Amplify.

Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console

Puoi scrivere ed eseguire codice per creare un URL che permette agli utenti che accedono alla rete dell'organizzazione di accedere in modo sicuro alla AWS Management Console. L'URL include un token di accesso che viene inviato AWS e utilizzato per autenticare l'utente. AWS La sessione console risultante potrebbe includere una distinta `AccessKeyId` a causa della federazione. Per tracciare l'utilizzo delle chiavi di accesso per l'accesso alla federazione tramite CloudTrail eventi correlati, vedi Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail e accedi agli eventi.AWS Management Console

> **Note**
>
> Se la tua organizzazione usa un provider di identità (IdP) compatibile con SAML, puoi configurare l'accesso alla console senza la necessità di scrivere codice. Ciò è possibile con provider come Microsoft Active Directory Federation Services oppure Shibboleth open source. Per informazioni dettagliate, vedi Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console.

Per consentire agli utenti dell'organizzazione di accedere a AWS Management Console, è possibile creare un broker di identità personalizzato che esegua i seguenti passaggi:

1. Verifica che l'utente sia autenticato dal sistema di identità locale.
2. Chiamate le operazioni AWS Security Token Service AssumeRole(AWS STS) (consigliato) o GetFederationTokenAPI per ottenere credenziali di sicurezza temporanee per l'utente. Per informazioni sui diversi metodi che si possono utilizzare per assumere un ruolo, consulta Utilizzo di ruoli IAM. Per informazioni su come passare i tag di sessione facoltativi quando si ottengono le credenziali di sicurezza, consulta Passare i tag di sessione AWS STS.
   - Se usi una delle operazioni API `AssumeRole*` per ottenere credenziali di sicurezza temporanee per un ruolo, puoi includere il parametro `DurationSeconds` nella chiamata. Questo parametro specifica la durata della sessione del ruolo, da 900 secondi (15 minuti) fino all'impostazione di durata massima della sessione per il ruolo. Quando si utilizza `DurationSeconds` in un'operazione `AssumeRole*`, è necessario chiamarlo come un utente IAM con credenziali a lungo termine. In caso contrario, la chiamata all'endpoint di federazione nella fase 3 ha esito negativo. Per informazioni su come visualizzare o modificare il valore massimo per un ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.
   - Se usi l'operazione API `GetFederationToken` per ottenere le credenziali, puoi includere il parametro `DurationSeconds` nella chiamata. Questo parametro specifica la durata della sessione del ruolo. Il valore può variare da 900 secondi (15 minuti) a 129.600 secondi (36 ore). Puoi effettuare questa chiamata API solo utilizzando le credenziali di AWS sicurezza a lungo termine di un utente IAM. Puoi anche effettuare queste chiamate utilizzando Utente root dell'account AWS le credenziali, ma non è consigliabile. Se effettui la chiamata come utente root, la sessione di default dura un'ora. In alternativa, puoi specificare una sessione con durata compresa tra 900 secondi (15 minuti) e 3.600 secondi (un'ora).
3. Chiama l'endpoint AWS della federazione e fornisci le credenziali di sicurezza temporanee per richiedere un token di accesso.
4. Crea un URL per la console che include il token:
   - Se usi una delle operazioni API `AssumeRole*` nell'URL, puoi includere il parametro HTTP `SessionDuration`. Questo parametro specifica la durata della sessione della console, da 900 secondi (15 minuti) a 43.200 secondi (12 ore).
   - Se usi l'operazione API `GetFederationToken` nell'URL, puoi includere il parametro `DurationSeconds`. Questo parametro specifica la durata della sessione della console federata. Il valore può variare da 900 secondi (15 minuti) a 129.600 secondi (36 ore).

> **Note**
>
> - Non usare il parametro HTTP `SessionDuration` se hai ottenuto le credenziali temporanee con `GetFederationToken`. Ciò causerebbe un errore dell'operazione.
> - L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, le nuove credenziali sono limitate a una durata massima di un'ora. Quando si utilizzano i ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2, tali applicazioni non sono soggette a questa limitazione.

5. Fornisce l'URL all'utente o richiama l'URL per conto dell'utente.

L'URL fornito dall'endpoint di federazione è valido per 15 minuti dopo la creazione. Questo intervallo di tempo è diverso dalla durata (in secondi) della sessione delle credenziali di sicurezza temporanee associata all'URL. Queste credenziali sono valide per la durata specificata al momento della creazione, a partire dal momento in cui sono state create.

> **Important**
>
> L'URL concede l'accesso alle tue AWS risorse tramite, AWS Management Console se hai abilitato le autorizzazioni nelle credenziali di sicurezza temporanee associate. Per questo motivo, devi trattare l'URL come segreto. Ti consigliamo di restituire l'URL attraverso un reindirizzamento sicuro, ad esempio usando un codice di stato della risposta HTTP 302 in una connessione SSL. Per ulteriori informazioni sul codice di stato della risposta HTTP 302, consulta RFC 2616, sezione 10.3.3.

Per completare queste attività, puoi utilizzare l'API di query HTTPS per AWS Identity and Access Management (IAM) e AWS Security Token Service (AWS STS). In alternativa, puoi utilizzare linguaggi di programmazione come Java, Ruby o C # con l'SDK AWS appropriato. Ognuno di questi metodi è descritto negli argomenti seguenti.

Argomenti

- Codice di esempio per l'uso delle operazioni API di query IAM
- Codice di esempio con Python
- Esempio di codice con Java

- Esempio ci creazione dell'URL (Ruby)

Codice di esempio per l'uso delle operazioni API di query IAM

Puoi creare un URL che fornisca agli utenti federati l'accesso diretto a. AWS Management Console Questa attività utilizza le API di interrogazione IAM e AWS STS HTTPS. Per ulteriori informazioni su come effettuare richieste di query, consulta Effettuare richieste di query.

> **Note**
>
> La procedura seguente contiene esempi di stringhe di testo. Per migliorare la leggibilità, sono state aggiunte interruzioni di riga in alcuni degli esempi più lunghi. Quando crei queste stringhe per l'uso, ometti le interruzioni di riga.

Per consentire a un utente federato di accedere alle tue risorse dal AWS Management Console

1. Autentica l'utente nel sistema di identità e autorizzazione.
2. Ottieni credenziali di sicurezza temporanee per l'utente. Le credenziali temporanee sono costituite da un ID chiave di accesso, una chiave di accesso segreta e un token di sessione. Per ulteriori informazioni sulla creazione di credenziali temporanee, consulta Credenziali di sicurezza temporanee in IAM.

   Per ottenere credenziali temporanee, chiamate l' AWS STS AssumeRoleAPI (scelta consigliata) o l'GetFederationTokenAPI. Per ulteriori informazioni sulle differenze tra queste operazioni API, consulta Comprendere le opzioni API per delegare in modo sicuro l'accesso all' AWS account nel blog sulla AWS sicurezza.

   > **Important**
   >
   > Quando utilizzi l'GetFederationTokenAPI per creare credenziali di sicurezza temporanee, devi specificare le autorizzazioni che le credenziali concedono all'utente che assume il ruolo. Per le operazioni API che iniziano con `AssumeRole*`, è necessario usare un ruolo IAM per assegnare le autorizzazioni. Per le altre operazioni API, il meccanismo varia a seconda dell'API. Per ulteriori dettagli, consulta Controllo delle autorizzazioni per le credenziali di sicurezza temporanee. Inoltre, se usi le operazioni API `AssumeRole*`,

devi chiamarle come utente IAM con credenziali a lungo termine. In caso contrario, la chiamata all'endpoint di federazione nella fase 3 ha esito negativo.

3. Dopo aver ottenuto le credenziali di sicurezza temporanee, integrale in una stringa di sessione JSON per scambiarle con un token di accesso. Nell'esempio seguente viene illustrato come codificare le credenziali. Sostituisci il testo segnaposto con i valori appropriati delle credenziali ricevute nella fase precedente.

```
{"sessionId":"*** temporary access key ID ***",
"sessionKey":"*** temporary secret access key ***",
"sessionToken":"*** session token ***"}
```

4. Effettuare la codifica tramite URL della stringa di sessione della fase precedente. Poiché le informazioni codificate sono informazioni sensibili, ti consigliamo di evitare l'uso di un servizio Web per la codifica. Usa invece una funzione o una caratteristica installata in locale nel kit di strumenti di sviluppo per codificare queste informazioni in modo sicuro. Puoi usare la funzione `urllib.quote_plus` in Python, la funzione `URLEncoder.encode` in Java o la funzione `CGI.escape` in Ruby. Consulta gli esempi più avanti in questo argomento.

5.

> **Note**
>
> AWS supporta le richieste POST qui.

Invia la tua richiesta all'endpoint della AWS federazione:

`https://region-code.signin.aws.amazon.com/federation`

Per un elenco dei possibili valori di *region-code*, consulta la colonna Region (Regione) in Endpoint di accesso AWS. Facoltativamente, puoi utilizzare l'endpoint federativo di AWS accesso predefinito:

`https://signin.aws.amazon.com/federation`

La richiesta deve includere i parametri `Action` e `Session` e, facoltativamente, se è stata utilizzata un'operazione API `AssumeRole*`, un parametro HTTP `SessionDuration` come illustrato nell'esempio seguente.

```
Action = getSigninToken
SessionDuration = time in seconds
```

```
Session = *** the URL encoded JSON string created in steps 3 & 4 ***
```

> **Note**
>
> Le seguenti istruzioni in questa fase funzionano solo con le richieste GET.

Il parametro HTTP `SessionDuration` specifica la durata della sessione della console. Si tratta di un valore diverso rispetto alla durata delle credenziali temporanee specificato usando il parametro `DurationSeconds`. Puoi specificare un valore massimo di `SessionDuration` pari a 43.200 (12 ore). Se il `SessionDuration` parametro non è presente, la sessione utilizza per impostazione predefinita la durata delle credenziali recuperate AWS STS nel passaggio 2 (che per impostazione predefinita è un'ora). Consultare la documentazione per l'API `AssumeRole` per informazioni dettagliate su come specificare una durata tramite il parametro `DurationSeconds`. La possibilità di creare una sessione della console più lunga di un'ora è intrinseca nell'operazione `getSigninToken` dell'endpoint di federazione.

> **Note**
>
> - Non usare il parametro HTTP `SessionDuration` se hai ottenuto le credenziali temporanee con `GetFederationToken`. Ciò causerebbe un errore dell'operazione.
> - L'utilizzo delle credenziali perché un ruolo assuma un ruolo diverso viene chiamato concatenamento dei ruoli. Quando si utilizza il concatenamento dei ruoli, le nuove credenziali sono limitate a una durata massima di un'ora. Quando si utilizzano i ruoli per concedere autorizzazioni alle applicazioni eseguite su istanze EC2, tali applicazioni non sono soggette a questa limitazione.

Quando abiliti le sessioni della console con una durata estesa, aumenti il rischio di esposizione delle credenziali. Per mitigare questo rischio, è possibile disabilitare immediatamente le sessioni della console attive per tutti i ruoli, scegliendo Revoca sessioni nella pagina Riepilogo ruolo nella console IAM. Per ulteriori informazioni, consulta Revoca delle credenziali di sicurezza temporanee per i ruoli IAM.

Di seguito è riportato un esempio di richiesta. Per le righe è impostato il ritorno a capo per semplificare la lettura, ma devi inviare la richiesta come stringa su un'unica riga.

```
https://signin.aws.amazon.com/federation
?Action=getSigninToken
&SessionDuration=1800
&Session=%7B%22sessionId%22%3A+%22ASIAJUMHIZPTOKTBMK5A%22%2C+%22sessionKey%22
%3A+%22LSD7LWI%2FL%2FN%2BgYpan5QFz0XUpc8s7HYjRsgcsrsm%22%2C+%22sessionToken%2
2%3A+%22FQoDYXdzEBQaDLbj3VWv2u50NN%2F3yyLSASwYtWhPnGPMNmzZFfZsL0Qd3vtYHw5A5dW
AjOsrkdPkghomIe3mJip5%2F0djDBbo7SmO%2FENDEiCdpsQKodTpleKA8xQq0CwFg6a69xdEBQT8
FipATnLbKoyS4b%2FebhnsTUjZZQWp0wXXqFF7gSm%2FMe2tXe0jzsdP0O12obez9lijPSdF1k2b5
PfGhiuyAR9aD5%2BubM0pY86fKex1qsytjvyTbZ9nXe6DvxVDcnCOhOGETJ7XFkSFdH0v%2FYR25C
UAhJ3nXIkIbG7Ucv9cOEpCf%2Fg23ijRgILIBQ%3D%3D%22%7D
```

La risposta dell'endpoint di federazione è un documento JSON con un valore `SigninToken`. L'aspetto sarà simile all'esempio seguente.

```
{"SigninToken":"*** the SigninToken string ***"}
```

6.

> **Note**
>
> AWS supporta le richieste POST qui.

Infine, crea l'URL che gli utenti federati possono usare per accedere alla AWS Management Console. L'URL corrisponde all'URL dell'endpoint di federazione usato in Step 5, con l'aggiunta dei parametri seguenti:

```
?Action = login
&Issuer = *** the form-urlencoded URL for your internal sign-in page ***
&Destination = *** the form-urlencoded URL to the desired AWS console page ***
&SigninToken = *** the value of SigninToken received in the previous step ***
```

> **Note**
>
> Le seguenti istruzioni in questa fase funzionano solo con utilizzando l'API GET.

L'esempio seguente mostra l'aspetto dell'URL finale. L'URL è valido per 15 minuti dal momento della creazione. Le credenziali di sicurezza temporanee e la sessione della console incorporate

nell'URL sono valide per la durata specificata nel parametro HTTP `SessionDuration` al momento della richiesta iniziale.

```
https://signin.aws.amazon.com/federation
?Action=login
&Issuer=https%3A%2F%2Fexample.com
&Destination=https%3A%2F%2Fconsole.aws.amazon.com%2F
&SigninToken=VCQgs5qZZt3Q6fn8Tr5EXAMPLEmLnwB7JjUc-SHwnUUWabcRdnWsi4DBn-dvC
CZ85wrD0nmldUcZEXAMPLE-vXYH4Q__mleuF_W2BE5HYexbe9y4Of-kje53SsjNNecATfjIzpW1
WibbnH6YcYRiBoffZBGExbEXAMPLE5aiKX4THWjQKC6gg6alHu6JFrnOJoK3dtP6I9a6hi6yPgm
iOkPZMmNGmhsvVxetKzr8mx3pxhHbMEXAMPLETv1pij0rok3IyCR2YVcIjqwfWv32HU2Xlj471u
3fU6uOfUComeKiqTGX974xzJOZbdmX_t_lLrhEXAMPLEDDIisSnyHGw2xaZZqudm4mo2uTDk9Pv
9l5K0ZCqIgEXAMPLEcA6tgLPykEWGUyH6BdSC6166n4M4JkXIQgac7_7821YqixsNxZ6rsrpzwf
nQoS14O7R0eJCCJ684EXAMPLEZRdBNnuLbUYpz2Iw3vIN0tQgOujwnwydPscM9F7foaEK3jwMkg
Apeb1-6L_OB12MZhuFxx55555EXAMPLEhyETEd4ZulKPdXHkgl6T9ZkIlHz2Uy1RUTUhhUxNtSQ
nWc5xkbBoEcXqpoSIeK7yhje9Vzhd61AEXAMPLElbWeouACEMG6-Vd3dAgFYd6i5FYoyFrZLWvm
0LSG7RyYKeYN5VIzUk3YWQpyjP0RiT5KUrsUi-NEXAMPLExMOMdoODBEgKQsk-iu2ozh6r8bxwC
RNhujg
```

Codice di esempio con Python

Gli esempi seguenti mostrano come utilizzare Python a livello di programmazione per formulare un URL che conceda agli utenti federati l'accesso diretto alla AWS Management Console. Gli esempi sono due:

- Federa tramite richieste GET a AWS
- Federate tramite richieste POST a AWS

Entrambi gli esempi utilizzano l'AssumeRoleAPI AWS SDK for Python (Boto3)and per ottenere credenziali di sicurezza temporanee.

Utilizzo delle richieste GET

```
import urllib, json, sys
import requests # 'pip install requests'
import boto3 # AWS SDK for Python (Boto3) 'pip install boto3'

# Step 1: Authenticate user in your own identity system.

# Step 2: Using the access keys for an IAM user in your Account AWS,
```

```
# call "AssumeRole" to get temporary access keys for the federated user

# Note: Calls to AWS STS AssumeRole must be signed using the access key ID
# and secret access key of an IAM user or using existing temporary credentials.
# The credentials can be in Amazon EC2 instance metadata, in environment variables,
# or in a configuration file, and will be discovered automatically by the
# client('sts') function. For more information, see the Python SDK docs:
# http://boto3.readthedocs.io/en/latest/reference/services/sts.html
# http://boto3.readthedocs.io/en/latest/reference/services/
sts.html#STS.Client.assume_role
sts_connection = boto3.client('sts')

assumed_role_object = sts_connection.assume_role(
    RoleArn="arn:aws:iam::account-id:role/ROLE-NAME",
    RoleSessionName="AssumeRoleSession",
)

# Step 3: Format resulting temporary credentials into JSON
url_credentials = {}
url_credentials['sessionId'] =
 assumed_role_object.get('Credentials').get('AccessKeyId')
url_credentials['sessionKey'] =
 assumed_role_object.get('Credentials').get('SecretAccessKey')
url_credentials['sessionToken'] =
 assumed_role_object.get('Credentials').get('SessionToken')
json_string_with_temp_credentials = json.dumps(url_credentials)

# Step 4. Make request to AWS federation endpoint to get sign-in token. Construct the
 parameter string with
# the sign-in action request, a 12-hour session duration, and the JSON document with
 temporary credentials
# as parameters.
request_parameters = "?Action=getSigninToken"
request_parameters += "&SessionDuration=43200"
if sys.version_info[0] < 3:
    def quote_plus_function(s):
        return urllib.quote_plus(s)
else:
    def quote_plus_function(s):
        return urllib.parse.quote_plus(s)
request_parameters += "&Session=" +
 quote_plus_function(json_string_with_temp_credentials)
request_url = "https://signin.aws.amazon.com/federation" + request_parameters
r = requests.get(request_url)
```

```
# Returns a JSON document with a single element named SigninToken.
signin_token = json.loads(r.text)

# Step 5: Create URL where users can use the sign-in token to sign in to
# the console. This URL must be used within 15 minutes after the
# sign-in token was issued.
request_parameters = "?Action=login"
request_parameters += "&Issuer=Example.org"
request_parameters += "&Destination=" + quote_plus_function("https://
console.aws.amazon.com/")
request_parameters += "&SigninToken=" + signin_token["SigninToken"]
request_url = "https://signin.aws.amazon.com/federation" + request_parameters

# Send final URL to stdout
print (request_url)
```

Utilizzo delle richieste POST

```
import urllib, json, sys
import requests # 'pip install requests'
import boto3 # AWS SDK for Python (Boto3) 'pip install boto3'
import os
from selenium import webdriver # 'pip install selenium', 'brew install chromedriver'

# Step 1: Authenticate user in your own identity system.

# Step 2: Using the access keys for an IAM user in your A Account AWS,
# call "AssumeRole" to get temporary access keys for the federated user

# Note: Calls to AWS STS AssumeRole must be signed using the access key ID
# and secret access key of an IAM user or using existing temporary credentials.
# The credentials can be in Amazon EC2 instance metadata, in environment variables,

# or in a configuration file, and will be discovered automatically by the
# client('sts') function. For more information, see the Python SDK docs:
# http://boto3.readthedocs.io/en/latest/reference/services/sts.html
# http://boto3.readthedocs.io/en/latest/reference/services/
sts.html#STS.Client.assume_role
if sys.version_info[0] < 3:
    def quote_plus_function(s):
        return urllib.quote_plus(s)
else:
    def quote_plus_function(s):
```

```
        return urllib.parse.quote_plus(s)

sts_connection = boto3.client('sts')

assumed_role_object = sts_connection.assume_role(
    RoleArn="arn:aws:iam::account-id:role/ROLE-NAME",
    RoleSessionName="AssumeRoleDemoSession",
)

# Step 3: Format resulting temporary credentials into JSON
url_credentials = {}
url_credentials['sessionId'] =
 assumed_role_object.get('Credentials').get('AccessKeyId')
url_credentials['sessionKey'] =
 assumed_role_object.get('Credentials').get('SecretAccessKey')
url_credentials['sessionToken'] =
 assumed_role_object.get('Credentials').get('SessionToken')
json_string_with_temp_credentials = json.dumps(url_credentials)

# Step 4. Make request to AWS federation endpoint to get sign-in token. Construct the
 parameter string with
# the sign-in action request, a 12-hour session duration, and the JSON document with
 temporary credentials
# as parameters.
request_parameters = {}
request_parameters['Action'] = 'getSigninToken'
request_parameters['SessionDuration'] = '43200'
request_parameters['Session'] = json_string_with_temp_credentials

request_url = "https://signin.aws.amazon.com/federation"
r = requests.post( request_url, data=request_parameters)

# Returns a JSON document with a single element named SigninToken.
signin_token = json.loads(r.text)

# Step 5: Create a POST request where users can use the sign-in token to sign in to
# the console. The POST request must be made within 15 minutes after the
# sign-in token was issued.
request_parameters = {}
request_parameters['Action'] = 'login'
request_parameters['Issuer']='Example.org'
request_parameters['Destination'] = 'https://console.aws.amazon.com/'
request_parameters['SigninToken'] =signin_token['SigninToken']
```

```
jsrequest = '''
var form = document.createElement('form');
form.method = 'POST';
form.action = '{request_url}';
request_parameters = {request_parameters}
for (var param in request_parameters) {{
    if (request_parameters.hasOwnProperty(param)) {{
        const hiddenField = document.createElement('input');
        hiddenField.type = 'hidden';
        hiddenField.name = param;
        hiddenField.value = request_parameters[param];
        form.appendChild(hiddenField);
    }}
}}
document.body.appendChild(form);
form.submit();
'''.format(request_url=request_url, request_parameters=request_parameters)

driver = webdriver.Chrome()
driver.execute_script(jsrequest);
```

Esempio di codice con Java

L'esempio seguente mostra come usare Java a livello di programmazione per creare un URL che concede agli utenti federati l'accesso diretto alla AWS Management Console. Nel frammento di codice seguente viene utilizzato AWS SDK for Java.

```
import java.net.URLEncoder;
import java.net.URL;
import java.net.URLConnection;
import java.io.BufferedReader;
import java.io.InputStreamReader;
// Available at http://www.json.org/java/index.html
import org.json.JSONObject;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClient;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;


/* Calls to AWS STS API operations must be signed using the access key ID
```

```
   and secret access key of an IAM user or using existing temporary
   credentials. The credentials should not be embedded in code. For
   this example, the code looks for the credentials in a
   standard configuration file.
*/
AWSCredentials credentials =
  new PropertiesCredentials(
         AwsConsoleApp.class.getResourceAsStream("AwsCredentials.properties"));

AWSSecurityTokenServiceClient stsClient =
  new AWSSecurityTokenServiceClient(credentials);

GetFederationTokenRequest getFederationTokenRequest =
  new GetFederationTokenRequest();
getFederationTokenRequest.setDurationSeconds(1800);
getFederationTokenRequest.setName("UserName");

// A sample policy for accessing Amazon Simple Notification Service (Amazon SNS) in the
 console.

String policy = "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Action\":\"sns:*\"," +
  "\"Effect\":\"Allow\",\"Resource\":\"*\"}]}";

getFederationTokenRequest.setPolicy(policy);

GetFederationTokenResult federationTokenResult =
  stsClient.getFederationToken(getFederationTokenRequest);

Credentials federatedCredentials = federationTokenResult.getCredentials();

// The issuer parameter specifies your internal sign-in
// page, for example https://mysignin.internal.mycompany.com/.
// The console parameter specifies the URL to the destination console of the
// AWS Management Console. This example goes to Amazon SNS.
// The signin parameter is the URL to send the request to.

String issuerURL = "https://mysignin.internal.mycompany.com/";
String consoleURL = "https://console.aws.amazon.com/sns";
String signInURL = "https://signin.aws.amazon.com/federation";

// Create the sign-in token using temporary credentials,
// including the access key ID,  secret access key, and session token.
String sessionJson = String.format(
  "{\"%1$s\":\"%2$s\",\"%3$s\":\"%4$s\",\"%5$s\":\"%6$s\"}",
```

```
  "sessionId", federatedCredentials.getAccessKeyId(),
  "sessionKey", federatedCredentials.getSecretAccessKey(),
  "sessionToken", federatedCredentials.getSessionToken());

// Construct the sign-in request with the request sign-in token action, a
// 12-hour console session duration, and the JSON document with temporary
// credentials as parameters.

String getSigninTokenURL = signInURL +
                           "?Action=getSigninToken" +
                           "&DurationSeconds=43200" +
                           "&SessionType=json&Session=" +
                           URLEncoder.encode(sessionJson,"UTF-8");

URL url = new URL(getSigninTokenURL);

// Send the request to the AWS federation endpoint to get the sign-in token
URLConnection conn = url.openConnection ();

BufferedReader bufferReader = new BufferedReader(new
  InputStreamReader(conn.getInputStream()));
String returnContent = bufferReader.readLine();

String signinToken = new JSONObject(returnContent).getString("SigninToken");

String signinTokenParameter = "&SigninToken=" + URLEncoder.encode(signinToken,"UTF-8");

// The issuer parameter is optional, but recommended. Use it to direct users
// to your sign-in page when their session expires.

String issuerParameter = "&Issuer=" + URLEncoder.encode(issuerURL, "UTF-8");

// Finally, present the completed URL for the AWS console session to the user

String destinationParameter = "&Destination=" + URLEncoder.encode(consoleURL,"UTF-8");
String loginURL = signInURL + "?Action=login" +
                     signinTokenParameter + issuerParameter + destinationParameter;
```

Esempio ci creazione dell'URL (Ruby)

L'esempio seguente mostra come usare Ruby a livello di programmazione per creare un URL che concede agli utenti federati l'accesso diretto alla AWS Management Console. In questo frammento di codice viene utilizzato AWS SDK for Ruby.

```
require 'rubygems'
require 'json'
require 'open-uri'
require 'cgi'
require 'aws-sdk'

# Create a new STS instance
#
# Note: Calls to AWS STS API operations must be signed using an access key ID
# and secret access key. The credentials can be in EC2 instance metadata
# or in environment variables and will be automatically discovered by
# the default credentials provider in the AWS Ruby SDK.
sts = Aws::STS::Client.new()

# The following call creates a temporary session that returns
# temporary security credentials and a session token.
# The policy grants permissions to work
# in the AWS SNS console.

session = sts.get_federation_token({
  duration_seconds: 1800,
  name: "UserName",
  policy: "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",\"Action\":
\"sns:*\",\"Resource\":\"*\"}}",
})

# The issuer value is the URL where users are directed (such as
# to your internal sign-in page) when their session expires.
#
# The console value specifies the URL to the destination console.
# This example goes to the Amazon SNS console.
#
# The sign-in value is the URL of the AWS STS federation endpoint.
issuer_url = "https://mysignin.internal.mycompany.com/"
console_url = "https://console.aws.amazon.com/sns"
signin_url = "https://signin.aws.amazon.com/federation"

# Create a block of JSON that contains the temporary credentials
# (including the access key ID, secret access key, and session token).
session_json = {
  :sessionId => session.credentials[:access_key_id],
  :sessionKey => session.credentials[:secret_access_key],
  :sessionToken => session.credentials[:session_token]
```

```
}.to_json

# Call the federation endpoint, passing the parameters
# created earlier and the session information as a JSON block.
# The request returns a sign-in token that's valid for 15 minutes.
# Signing in to the console with the token creates a session
# that is valid for 12 hours.
get_signin_token_url = signin_url +
                       "?Action=getSigninToken" +
                       "&SessionType=json&Session=" +
                       CGI.escape(session_json)

returned_content = URI.parse(get_signin_token_url).read

# Extract the sign-in token from the information returned
# by the federation endpoint.
signin_token = JSON.parse(returned_content)['SigninToken']
signin_token_param = "&SigninToken=" + CGI.escape(signin_token)

# Create the URL to give to the user, which includes the
# sign-in token and the URL of the console to open.
# The "issuer" parameter is optional but recommended.
issuer_param = "&Issuer=" + CGI.escape(issuer_url)
destination_param = "&Destination=" + CGI.escape(console_url)
login_url = signin_url + "?Action=login" + signin_token_param +
  issuer_param + destination_param
```

Risorse aggiuntive per le credenziali di sicurezza temporanee

I seguenti scenari e applicazioni possono essere utili per l'utilizzo di credenziali di sicurezza temporanee:

- Come integrarsi AWS STS SourceIdentity con il tuo provider di identità. Questo post mostra come configurare l' AWS STS `SourceIdentity`attributo quando usi Okta, Ping o OneLogin come IdP.
- Federazione OIDC. Questa sezione illustra come configurare i ruoli IAM quando si utilizza la federazione OIDC e l'API. `AssumeRoleWithWebIdentity`
- Configurazione dell'accesso alle API protetto da MFA. In questo argomento viene descritto come utilizzare i ruoli per richiedere l'autenticazione a più fattori (MFA) per proteggere le operazioni API sensibili nel tuo account.

Per ulteriori informazioni sulle politiche e le autorizzazioni, AWS consulta i seguenti argomenti:

- Gestione degli accessi AWS alle risorse
- Logica di valutazione delle policy.
- Gestione delle autorizzazioni di accesso alle risorse di Amazon S3 in Guida per l'utente di Amazon Simple Storage Service.
- Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Tagging delle risorse IAM

Un tag è un'etichetta di attributi personalizzata assegnata a una risorsa AWS . Ogni tag è costituito da due parti:

- Una chiave di tag (ad esempio, `CostCenter`, `Environment`, `Project` o `Purpose`).
- Un campo facoltativo noto come valore del tag (ad esempio, `111122223333`, `Production` o un nome di team). Non specificare il valore del tag equivale a utilizzare una stringa vuota.

Tutti questi sono noti come coppie chiave-valore. Per i limiti sul numero di tag che è possibile avere sulle risorse IAM, consulta IAM e AWS STS quote.

> **Note**
>
> Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

I tag ti aiutano a identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, è possibile assegnare lo stesso tag a un ruolo IAM che si assegna a un bucket Amazon S3. Per ulteriori informazioni sulle strategie di tagging, consulta la Guida per l'utente delle risorse di etichettatura AWS.

Oltre a identificare, organizzare e monitorare le risorse IAM con i tag, puoi usare i tag nelle policy IAM per controllare chi può visualizzare e interagire con le risorse. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Scegli una convenzione di denominazione dei AWS tag

Quando si inizia a collegare tag alle risorse IAM, scegli attentamente la convenzione di denominazione dei tag. Applica la stessa convenzione a tutti i AWS tag. Ciò è particolarmente importante se si utilizzano i tag nelle politiche per controllare l'accesso alle AWS risorse. Se utilizzi già tag in AWS, rivedi la convenzione di denominazione e modificala di conseguenza.

> **Note**
>
> Se il tuo account è membro di AWS Organizations, consulta le politiche sui tag nella guida per l'utente di Organizations per saperne di più sull'uso dei tag in Organizations.

Best practice per la denominazione dei tag

Di seguito sono riportate alcune best practice e convenzioni di denominazione per i tag.

Assicurati che i nomi dei tag vengano utilizzati in modo coerente. Ad esempio, i tag `CostCenter` e `costcenter` sono diversi, pertanto uno potrebbe essere configurato come tag di allocazione dei costi per l'analisi e il report finanziario mentre l'altro no. Allo stesso modo, il `Name` tag viene visualizzato nella AWS Console per molte risorse, ma il `name` tag no. Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

Alcuni tag sono predefiniti AWS o creati automaticamente da vari AWS servizi. Molti nomi AWS di tag definiti utilizzano solo lettere minuscole, con trattini che separano le parole nel nome e prefissi per identificare il servizio di origine del tag. Per esempio:

- `aws:ec2spot:fleet-request-id` identifica la richiesta di istanza Spot di Amazon EC2 che ha avviato l'istanza.
- `aws:cloudformation:stack-name`identifica lo stack che ha creato la risorsa. AWS CloudFormation
- `elasticbeanstalk:environment-name` identifica l'applicazione che ha creato la risorsa.

Prendi in considerazione la possibilità di assegnare un nome ai tag utilizzando tutte lettere minuscole, con trattini che separano le parole e un prefisso che identifichi il nome dell'organizzazione o il nome abbreviato. Ad esempio, per una società fittizia denominata AnyCompany, è possibile definire tag come:

- `anycompany:cost-center` per identificare il codice interno del centro di costo
- `anycompany:environment-type` per identificare se l'ambiente è in fase di sviluppo, test o produzione
- `anycompany:application-id` per identificare l'applicazione per cui è stata creata la risorsa

Il prefisso garantisce che i tag siano chiaramente identificati come definiti dall'organizzazione e non da AWS uno strumento di terze parti che potreste utilizzare. L'uso di tutte le lettere minuscole con i trattini per i separatori evita confusione su come scrivere il nome di un tag. Ad esempio, `anycompany:project-id` è più semplice da ricordare rispetto `ANYCOMPANY:ProjectID`, `anycompany:projectID` oppure `Anycompany:ProjectId`.

Regole per l'etichettatura in IAM e AWS STS

Un certo numero di convenzioni regola la creazione e l'applicazione di tag in IAM e AWS STS.

Denominazione di tag

Osserva le seguenti convenzioni quando formuli una convenzione di denominazione dei tag per risorse IAM, sessioni di assunzione di AWS STS ruoli e sessioni utente federate: AWS STS

Requisiti per i caratteri: chiavi e valori di tag possono includere qualsiasi combinazione di lettere, numeri, spazi e simboli _ . : / = + - @.

Distinzione tra maiuscole e minuscole: la distinzione tra maiuscole e minuscole per le chiavi di tag varia a seconda del tipo di risorsa IAM taggata. I valori chiave dei tag per utenti e ruoli IAM non distingue tra maiuscole e minuscole, anche se questi caratteri vengono mantenuti. Questo significa che non è possibile avere le chiavi di tag **`Department`** e **`department`** separate. Se hai assegnato a un utente il tag **`Department=finance`** e aggiungi il tag **`department=hr`**, quest'ultimo sostituirà il primo tag. Non viene aggiunto un secondo tag.

Per gli altri tipi di risorse IAM, i valori della chiave di tag fanno distinzione tra maiuscole e minuscole. Questo significa che è possibile avere chiavi di tag **`Costcenter`** e **`costcenter`** separate. Ad esempio, se sono stati applicati i tag a una policy gestita dal cliente con il tag **`Costcenter = 1234`** e si aggiunge il tag **`costcenter = 5678`**, il criterio avrà entrambe le chiavi di tag **`Costcenter`** e **`costcenter`**.

Come best practice, si consiglia di evitare l'uso di tag simili con un'applicazione di maiuscole e minuscole non coerente. Consigliamo di definire una strategia per l'uso delle lettere maiuscole e

minuscole nei tag e implementarla in modo coerente per tutti i tipi di risorse. Per ulteriori informazioni sulle migliori pratiche per l'etichettatura, consulta Tagging Resources in. AWS Riferimenti generali di AWS

Negli elenchi seguenti vengono illustrate le differenze nella distinzione tra maiuscole e minuscole per le chiavi di tag associate alle risorse IAM.

I valori delle chiavi tag non fanno distinzione tra maiuscole e minuscole:

- Ruoli IAM
- Utenti IAM

Le chiavi e i valori fanno distinzione tra maiuscole e minuscole.

- Policy gestite dal cliente
- Profili delle istanze
- Provider di identità OpenID Connect
- Provider di identità SAML
- Certificati server
- Dispositivi MFA virtuali

Inoltre, valgono le seguenti regole:

- Non è possibile creare una chiave o un valore di tag che inizi con il testo **`aws:`**. Questo prefisso di tag è riservato per AWS uso interno.
- È possibile creare un tag con un valore vuoto, ad esempio **`phoneNumber =`** . Non è possibile creare una chiave di tag vuota.
- Non è possibile specificare più valori in un singolo tag, ma è possibile creare una struttura multivalore personalizzata nel singolo valore. Ad esempio, supponiamo che l'utente Zhang lavori nel team di progettazione e nel team di QA. Se colleghi il tag **`team = Engineering`** e poi colleghi il tag **`team = QA`**, modifichi il valore del tag da **`Engineering`** a **`QA`**. Al contrario, è possibile includere più valori in un singolo tag con un separatore personalizzato. In questo esempio, è possibile collegare il tag **`team = Engineering:QA`** a Zhang.

> **Note**
>
> Per controllare l'accesso al team di progettazione in questo esempio utilizzando il tag **`team`**, è necessario creare una policy che consenta ogni configurazione che potrebbe includere **`Engineering`**, tra cui **`Engineering:QA`**. Per ulteriori informazioni sull'utilizzo dei tag nelle policy, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Applicazione e modifica di tag

Osserva le seguenti convenzioni quando applichi i tag alle risorse IAM:

- Puoi applicare tag alla maggior parte delle risorse IAM, ma non a gruppi, ruoli assunti, report di accesso o dispositivi MFA basati su hardware.
- Non è possibile utilizzare l'editor di tag per applicare i tag alle risorse IAM. L'editor di tag non supporta i tag IAM. Per ulteriori informazioni sull'utilizzo dell'editor di tag con altri servizi, consulta l'articolo relativo all'utilizzo dell'editor di tag nella Guida per l'utente della AWS Resource Groups .
- Per aggiungere i tag a una risorsa IAM, devi disporre di autorizzazioni specifiche. Per applicare o rimuovere i tag dalle risorse, è necessario disporre anche dell'autorizzazione per elencare i tag. Per ulteriori informazioni, consulta l'elenco degli argomenti relativi a ciascuna risorsa IAM alla fine di questa pagina.
- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.
- Puoi applicare lo stesso tag a più risorse IAM. Ad esempio, si supponga di avere un reparto denominato `AWS_Development` con 12 membri. È possibile avere 12 utenti e un ruolo con **`department`** come chiave del tag e **`awsDevelopment`** come valore (**`department = awsDevelopment`**). Puoi inoltre utilizzare lo stesso tag su risorse in altri servizi che supportano il tagging.
- Le entità IAM (utenti o ruoli) non possono avere più istanze della stessa chiave di tag. Ad esempio, se disponi di un utente con la coppia chiave-valore del tag **`costCenter = 1234`**, puoi collegare la coppia chiave-valore del tag **`costCenter = 5678`**. IAM aggiorna il valore del tag **`costCenter`** a **`5678`**.
- Per modificare un tag collegato a un'entità IAM (utente o ruolo), collega un tag con un nuovo valore per sovrascrivere il tag esistente. Ad esempio, supponiamo che tu disponga di un utente con la coppia chiave-valore del tag **`department = Engineering`**. Se devi spostare l'utente nel reparto

QA, puoi collegare la coppia chiave-valore del tag **`department = QA`** all'utente. In questo modo il valore **`Engineering`** della chiave di tag **`department`** viene sostituito con il valore **`QA`**.

Argomenti

Tagging di utenti IAM

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un utente IAM. Ad esempio, per aggiungere informazioni sulla posizione per un utente, puoi aggiungere la chiave di tag **`location`** e il valore di tag **`us_wa_seattle`**. In alternativa, puoi utilizzare tre coppie chiave-valore del tag per la posizione separate: **`loc-country = us`**, **`loc-state = wa`** e **`loc-city = seattle`**. È possibile usare i tag per controllare l'accesso di un utente alle risorse o per controllare quali tag possono essere collegati a un utente. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per il tagging degli utenti IAM

Per consentire a un utente IAM di aggiungere tag ad altri utenti, è necessario configurare le autorizzazioni relative. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListUserTags`
- `iam:TagUser`

- `iam:UntagUser`

Come consentire a un utente IAM di aggiungere, elencare o rimuovere un tag per un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'utente IAM che deve gestire i tag. Utilizza il tuo numero di account e sostituisci *<username>* con il nome dell'utente di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un utente IAM di gestire autonomamente i tag

Aggiungi l'istruzione seguente alla policy di autorizzazione per gli utenti per consentire loro di gestire i propri tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::user/${aws:username}"
}
```

Come consentire a un utente IAM di aggiungere un tag a un utente specifico

Aggiungi l'istruzione seguente alla policy delle autorizzazioni relativa all'utente IAM che potrà aggiungere ma non rimuovere i tag di un determinato utente.

> **Note**
>
> L'azione `iam:TagUser` richiede che tu includa anche l'azione `iam:ListUserTags`.

Per utilizzare questa policy, sostituisci *<username>* con il nome dell'utente di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag degli utenti IAM (console)

Puoi gestire i tag per gli utenti IAM dalla AWS Management Console.

Per gestire i tag degli utenti (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Users (Utenti) e selezionare il nome dell'utente da modificare.
3. Scegliere la scheda Tags (Tag) e completare una delle seguenti operazioni:
   - Scegli Add new tag (Aggiungi nuovo tag) se all'utente non sono ancora stati assegnati tag.
   - Scegli Manage tags (Gestisci tag) per gestire il set di tag esistente.
4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sugli utenti IAM (AWS CLI o AWS API)

Puoi elencare, collegare o rimuovere i tag per gli utenti IAM. Puoi utilizzare l'API AWS CLI o l' AWS API per gestire i tag per gli utenti IAM.

Per elencare i tag attualmente associati a un utente IAM (AWS CLI o AWS API)

- AWS CLI: era iam list-user-tags
- AWS API: ListUserTags

Per allegare tag a un utente IAM (AWS CLI o AWS API)

- AWS CLI: aws iam tag-user
- AWS API: TagUser

Per rimuovere i tag da un utente (AWS CLI o AWS API) IAM

- AWS CLI: aws iam untag-user
- AWS API: UntagUser

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Tagging di ruoli IAM

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un ruolo IAM. Ad esempio, per aggiungere informazioni sulla posizione per un ruolo, puoi aggiungere la chiave tag **`location`** e il valore tag **`us_wa_seattle`**. In alternativa, puoi utilizzare tre coppie chiave-valore del tag per la posizione separate: **`loc-country = us`**, **`loc-state = wa`** e **`loc-city = seattle`**. Puoi utilizzare i tag per controllare l'accesso di un ruolo alle risorse o per controllare quali tag possono essere collegati a un ruolo. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per il tagging dei ruoli IAM

Per permettere a un ruolo IAM di applicare tag ad altre entità (utenti o ruoli), devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListRoleTags`
- `iam:TagRole`
- `iam:UntagRole`
- `iam:ListUserTags`
- `iam:TagUser`
- `iam:UntagUser`

Come consentire a un ruolo IAM di aggiungere, elencare o rimuovere un tag per un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per il ruolo IAM che deve gestire i tag. Utilizza il tuo numero di account e sostituisci *<username>* con il nome dell'utente di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser",
        "iam:UntagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un ruolo IAM di aggiungere un tag a un utente specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione relativa al ruolo IAM che potrà aggiungere ma non rimuovere i tag di un utente specifico.

Per utilizzare questa policy, sostituisci *<username>* con il nome dell'utente di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListUserTags",
        "iam:TagUser"
    ],
    "Resource": "arn:aws:iam::<account-number>:user/<username>"
}
```

Come consentire a un ruolo IAM di aggiungere, elencare o rimuovere un tag per un ruolo specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per il ruolo IAM che deve gestire i tag. Sostituisci *<rolename>* con il nome del ruolo i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListRoleTags",
        "iam:TagRole",
        "iam:UntagRole"
    ],
    "Resource": "arn:aws:iam::<account-number>:role/<rolename>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui ruoli IAM (console)

Puoi gestire i tag per i ruoli IAM dalla AWS Management Console.

Per gestire i tag sui ruoli (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Roles (Ruoli) e selezionare il nome del ruolo che si desidera modificare.
3. Scegliere la scheda Tags (Tag) e completare una delle seguenti operazioni:

- Scegli Add new tag (Aggiungi nuovo tag) se al ruolo non sono ancora stati assegnati tag.
- Scegli Manage tags (Gestisci tag) per gestire il set di tag esistente.

4. Aggiungere o rimuovere i tag per completare il set di tag. Quindi, scegli Save changes (Salva modifiche).

Gestione dei tag sui ruoli (AWS CLI o AWS API) IAM

Puoi elencare, collegare o rimuovere i tag per i ruoli IAM. Puoi utilizzare AWS CLI o l' AWS API per gestire i tag per i ruoli IAM.

Per elencare i tag attualmente associati a un ruolo (AWS CLI o AWS API) IAM

- AWS CLI: era iam list-role-tags
- AWS API: ListRoleTags

Per allegare tag a un ruolo (AWS CLI o AWS API) IAM

- AWS CLI: aws iam tag-role
- AWS API: TagRole

Per rimuovere i tag da un ruolo (AWS CLI o AWS API) IAM

- AWS CLI: aws iam untag-role
- AWS API: UntagRole

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Modifica di politiche gestite dal cliente

È possibile utilizzare le coppie chiave-valore del tag IAM per aggiungere attributi personalizzati alle policy gestite dal cliente. Ad esempio, per applicare un tag a una policy con le informazioni sul reparto, puoi aggiungere la chiave tag **`Department`** e il valore tag **`eng`**. In alternativa, è

possibile contrassegnare i criteri per indicare che si riferiscono a un ambiente specifico, ad esempio **`Environment = lab`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a una risorsa. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per applicare i tag alle policy gestite dai clienti

È necessario configurare le autorizzazioni per consentire a un'entità IAM (utenti o ruoli) di applicare i tag alle policy gestite dal cliente. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListPolicyTags`
- `iam:TagPolicy`
- `iam:UntagPolicy`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per una policy gestita dal cliente

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Utilizza il tuo numero di account e sostituisci *`<policyname>`* con il nome della policy di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListPolicyTags",
        "iam:TagPolicy",
        "iam:UntagPolicy"
    ],
    "Resource": "arn:aws:iam::<account-number>:policy/<policyname>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a una determinata policy gestita dal cliente

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per una policy specifica.

> **Note**
>
> L'azione `iam:TagPolicy` richiede che tu includa anche l'azione `iam:ListPolicyTags`.

Per utilizzare questa policy, sostituisci *<policyname>* con il nome della policy di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListPolicyTags",
        "iam:TagPolicy"
    ],
    "Resource": "arn:aws:iam::<account-number>:policy/<policyname>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nelle policy gestiti dal cliente IAM (console)

Puoi gestire i tag delle policy gestite dal cliente IAM dalla AWS Management Console.

Per gestire i tag nelle policy gestite dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegliere Policies (Policy) e selezionare il nome della policy gestita dal cliente che si desidera modificare.
3. Scegli la scheda Tag, quindi scegli Gestisci tag.
4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sulle politiche gestite dai clienti (AWS CLI o AWS API) IAM

Puoi elencare, allegare o rimuovere i tag per le policy gestite dal cliente IAM. Puoi utilizzare l'API AWS CLI o l' AWS API per gestire i tag per le politiche gestite dai clienti IAM.

Per elencare i tag attualmente associati a una policy (AWS CLI o AWS API) gestita dai clienti IAM

- AWS CLI: era iam list-policy-tags
- AWS API: ListPolicyTags

Per allegare tag a una policy (AWS CLI o AWS API) gestita dai clienti IAM

- AWS CLI: aws iam tag-policy
- AWS API: TagPolicy

Per rimuovere i tag da una policy (AWS CLI o AWS API) gestita dai clienti IAM

- AWS CLI: aws iam untag-policy
- AWS API: UntagPolicy

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Tagging dei provider di identità IAM

Puoi utilizzare le coppie chiave-valore dei tag IAM per aggiungere attributi personalizzati ai provider di identità IAM ()IdPs.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Per ulteriori informazioni sull'etichettatura IdPs in IAM, consulta i seguenti argomenti:

Argomenti

- Applicazione di tag ai provider di identità OpenID Connect (OIDC)

- Tagging di provider di identità SAML per IAM

Applicazione di tag ai provider di identità OpenID Connect (OIDC)

Puoi utilizzare le coppie chiave-valore dei tag IAM per aggiungere attributi personalizzati ai provider di identità OpenID Connect (OIDC) IAM. Ad esempio, per identificare un provider di identità OIDC, puoi aggiungere la chiave tag **`google`** e il valore tag **`oidc`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Autorizzazioni necessarie per l'aggiunta di tag ai provider di identità OIDC IAM

Per permettere a un'entità IAM (utente o ruolo) di applicare tag ai provider di identità OIDC IAM, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListOpenIDConnectProviderTags`
- `iam:TagOpenIDConnectProvider`
- `iam:UntagOpenIDConnectProvider`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un provider di identità OIDC IAM

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituisci *<OIDC ProviderName >* con il nome del provider OIDC i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListOpenIDConnectProviderTags",
        "iam:TagOpenIDConnectProvider",
        "iam:UntagOpenIDConnectProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:oidc-provider/<OIDCProviderName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un provider di identità OIDC IAM specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato provider di identità.

> **Note**
>
> L'azione `iam:TagOpenIDConnectProvider` richiede che tu includa anche l'azione `iam:ListOpenIDConnectProviderTags`.

Per utilizzare questa politica, sostituisci *<OIDC ProviderName > con il nome del provider OIDC* i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListOpenIDConnectProviderTags",
        "iam:TagOpenIDConnectProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:oidc-provider/<OIDCProviderName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nei provider di identità OIDC IAM (console)

Puoi gestire i tag per i provider di identità OIDC IAM dalla AWS Management Console.

Per gestire i tag dei provider di identità OIDC (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Identity providers (Provider di identità), quindi scegli il nome del provider di identità che desideri aggiornare.
3. Nella sezione Tag (Tag) scegli Manage tags (Gestisci tag) e completa una delle seguenti azioni:

- Sceglie Add tag (Aggiungi tag) se il provider di identità OIDC non dispone ancora di tag o per aggiungere un nuovo tag.
- Modificare le chiavi e i valori dei tag esistenti.
- Per rimuovere un tag, scegli Remove tag (Rimuovi tag).

4. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sui provider di identità IAM OIDC (AWS CLI o AWS API)

Puoi elencare, allegare o rimuovere i tag per i provider di identità OIDC IAM. Puoi utilizzare l' AWS CLI o l' AWS API per gestire i tag per i provider di identità IAM OIDC.

Per elencare i tag attualmente associati a un provider di identità (o API) IAM OIDC AWS CLIAWS

- AWS CLI: aws iam -provider-tags list-open-id-connect
- AWS ListOpenAPI: ID ConnectProviderTags

Per allegare tag a un provider di identità (AWS CLI o AWS API) IAM OIDC

- AWS CLI: aws iam -provider tag-open-id-connect
- AWS API: ID TagOpen ConnectProvider

Per rimuovere i tag da un provider di identità (AWS CLI o AWS API) IAM OIDC

- AWS CLI: aws iam -provider untag-open-id-connect
- AWS API: ID UntagOpen ConnectProvider

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Tagging di provider di identità SAML per IAM

È possibile utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati ai provider di identità SAML. Ad esempio, per identificare un provider, puoi aggiungere la chiave tag

`okta` e il valore tag **`saml`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Autorizzazioni necessarie per l'assegnazione di tag ai provider di identità SAML

Devi configurare le autorizzazioni per consentire a un'entità IAM (utenti o ruoli) di taggare gli Identity Provider basati su SAML 2.0 (). IdPs Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListSAMLProviderTags`
- `iam:TagSAMLProvider`
- `iam:UntagSAMLProvider`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un provider di identità SAML

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituisci *<SAML ProviderName >* con il nome del provider SAML i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListSAMLProviderTags",
        "iam:TagSAMLProvider",
        "iam:UntagSAMLProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:saml-provider/<SAMLProviderName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un provider di identità SAML specifico

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato provider SAML.

> **Note**
>
> L'azione `iam:TagSAMLProvider` richiede che tu includa anche l'azione `iam:ListSAMLProviderTags`.

Per utilizzare questa politica, sostituisci *<SAML ProviderName >* con il nome del provider SAML i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListSAMLProviderTags",
        "iam:TagSAMLProvider"
    ],
    "Resource": "arn:aws:iam::<account-number>:saml-provider/<SAMLProviderName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag nei provider di identità SAML IAM (console)

È possibile gestire i tag per i provider di identità SAML IAM dalla AWS Management Console.

Per gestire i tag dei provider di identità SAML (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console, scegli Identity providers (Provider di identità), quindi scegli il nome del provider di identità SAML che desideri aggiornare.
3. Nella sezione Tag (Tag) scegli Manage tags (Gestisci tag) e completa una delle seguenti azioni:
   - Sceglie Add tag (Aggiungi tag) se il provider di identità SAML non dispone ancora di tag o per aggiungere un nuovo tag.
   - Modificare le chiavi e i valori dei tag esistenti.
   - Per rimuovere un tag, scegli Remove tag (Rimuovi tag).

4. Aggiungere o rimuovere i tag per completare il set di tag. Selezionare quindi Save changes (Salva modifiche).

Gestione dei tag sui provider di identità IAM SAML (AWS CLI o AWS API)

Puoi elencare, allegare o rimuovere i tag per i provider di identità SAML IAM. Puoi utilizzare l' AWS CLI o l' AWS API per gestire i tag per i provider di identità IAM SAML.

Per elencare i tag attualmente associati a un provider di identità SAML (AWS CLI o AWS API)

- AWS CLI: era iam list-saml-provider-tags
- AWS API: ListSAML ProviderTags

Per allegare tag a un provider di identità SAML (AWS CLI o API) AWS

- AWS CLI: era iam tag-saml-provider
- AWS API: TagSamlProvider

Per rimuovere i tag da un provider di identità SAML (o API)AWS CLIAWS

- AWS CLI: era iam untag-saml-provider
- AWS API: unTagSamlProvider

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Tagging dei profili dell'istanza per i ruoli Amazon EC2

Quando avvii un'istanza Amazon EC2, devi specificare un ruolo IAM da associare ad essa. Un profilo dell'istanza è un container per un ruolo IAM che puoi utilizzare per inoltrare informazioni sul ruolo a un'istanza Amazon EC2 quando questa viene avviata. Puoi etichettare i profili delle istanze quando usi l' AWS API AWS CLI o.

Puoi utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati a un profilo dell'istanza. Ad esempio, per aggiungere informazioni di reparto a un profilo di istanza, è possibile

aggiungere la chiave tag **`access-team`** e il valore tag **`eng`**. In questo modo i principali con tag corrispondenti possono accedere ai profili dell'istanza con lo stesso tag. È possibile utilizzare più coppie chiave-valore del tag per specificare un team e un progetto: **`access-team = eng`** e **`project = peg`**. È possibile usare i tag per controllare l'accesso di un utente alle risorse o per controllare quali tag possono essere collegati a un utente. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per il tagging dei profili dell'istanza

Per permettere a un'entità IAM (utente o ruolo) di aggiungere tag ai profili dell'istanza, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListInstanceProfileTags`
- `iam:TagInstanceProfile`
- `iam:UntagInstanceProfile`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un profilo dell'istanza

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituisci *< InstanceProfileName >* con il nome del profilo dell'istanza i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListInstanceProfileTags",
        "iam:TagInstanceProfile",
        "iam:UntagInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<account-number>:instance-profile/<InstanceProfileName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato profilo dell'istanza

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un determinato profilo dell'istanza.

> **Note**
>
> L'azione `iam:TagInstanceProfile` richiede che tu includa anche l'azione `iam:ListInstanceProfileTags`.

Per utilizzare questa politica, sostituisci *< InstanceProfileName >* con il nome del profilo dell'istanza i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListInstanceProfileTags",
        "iam:TagInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<account-number>:instance-profile/<InstanceProfileName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui profili di istanza (AWS CLI o AWS API)

È possibile elencare, allegare o rimuovere i tag dei profili dell'istanza. È possibile utilizzare l'API AWS CLI o l' AWS API per gestire i tag, ad esempio i profili.

Per elencare i tag attualmente associati a un profilo di istanza (AWS CLI o AWS API)

- AWS CLI: era iam list-instance-profile-tags
- AWS API: ListInstanceProfileTags

Per allegare tag a un profilo di istanza (AWS CLI o AWS API)

- AWS CLI: aws iam tag-instance-profile
- AWS API: TagInstanceProfile

Per rimuovere i tag da un profilo di istanza (AWS CLI o AWS API)

- AWS CLI: è stato io untag-instance-profile
- AWS API: UntagInstanceProfile

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Tagging dei certificati server

Se utilizzi IAM per gestire i certificati SSL/TLS, puoi etichettare i certificati server in IAM utilizzando l'API or. AWS CLI AWS Per i certificati in una regione supportata da AWS Certificate Manager (ACM), ti consigliamo di utilizzare ACM anziché IAM per fornire, gestire e distribuire i certificati del server. Nelle regioni non supportate, è necessario utilizzare IAM come gestore di certificati. Per informazioni sulle regioni supportate da ACM, consulta Endpoint e quote di AWS Certificate Manager nella Riferimenti generali di AWS.

Puoi utilizzare coppie chiave-valore del tag IAM per aggiungere attributi personalizzati a un certificato server. Ad esempio, per aggiungere informazioni sul proprietario o sull'amministratore di un certificato server, aggiungi la chiave tag **`owner`** e il valore tag **`net-eng`**. In alternativa, puoi specificare un centro di costo aggiungendo la chiave tag **`CostCenter`** e il valore tag **`1234`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati alle risorse. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per l'assegnazione di tag ai certificati server

Per permettere a un'entità IAM (utente o ruolo) di aggiungere i tag ai certificati server, devi configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListServerCertificateTags`
- `iam:TagServerCertificate`
- `iam:UntagServerCertificate`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un certificato server

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Usa il tuo numero di account e sostituisci *< CertificateName >* con il nome del certificato del server i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListServerCertificateTags",
        "iam:TagServerCertificate",
        "iam:UntagServerCertificate"
    ],
    "Resource": "arn:aws:iam::<account-number>:server-certificate/<CertificateName>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato certificato server

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve aggiungere, ma non rimuovere, i tag per un certificato server specifico.

> **Note**
>
> L'azione `iam:TagServerCertificate` richiede che tu includa anche l'azione `iam:ListServerCertificateTags`.

Per utilizzare questa politica, sostituisci *< CertificateName >* con il nome del certificato del server i cui tag devono essere gestiti. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListServerCertificateTags",
        "iam:TagServerCertificate"
    ],
    "Resource": "arn:aws:iam::<account-number>:server-certificate/<CertificateName>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag sui certificati del server (AWS CLI o AWS API)

È possibile elencare, allegare o rimuovere i certificati server. È possibile utilizzare l'API AWS CLI o l' AWS API per gestire i tag per i certificati del server.

Per elencare i tag attualmente collegati a un certificato del server (AWS CLI o AWS API)

- AWS CLI: era iam list-server-certificate-tags
- AWS API: ListServerCertificateTags

Per allegare tag a un certificato del server (AWS CLI o AWS API)

- AWS CLI: aws aim tag-server-certificate
- AWS API: TagServerCertificate

Per rimuovere i tag da un certificato del server (AWS CLI o AWS API)

- AWS CLI: aws aim untag-server-certificate
- AWS API: UntagServerCertificate

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Aggiunta di tag ai dispositivi MFA virtuali

Puoi utilizzare coppie chiave-valore di tag IAM per aggiungere attributi personalizzati a un dispositivo MFA virtuale. Ad esempio, per aggiungere informazioni sul centro di costo per il dispositivo MFA virtuale di un utente, puoi aggiungere il tag con chiave **`CostCenter`** e il tag con valore **`1234`**. Puoi usare i tag per controllare l'accesso alle risorse o per controllare quali tag possono essere collegati a un oggetto. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Puoi anche utilizzare i tag AWS STS per aggiungere attributi personalizzati quando assumi un ruolo o federi un utente. Per ulteriori informazioni, consulta Passare i tag di sessione AWS STS.

Autorizzazioni necessarie per la gestione dei tag dei dispositivi MFA virtuali

Per consentire a un'entità IAM (utente o ruolo) di aggiungere i tag ai dispositivi MFA virtuali, è necessario configurare le autorizzazioni. Puoi specificare una o tutte le seguenti operazioni del tag IAM in una policy IAM:

- `iam:ListMFADeviceTags`
- `iam:TagMFADevice`
- `iam:UntagMFADevice`

Come consentire a un'entità IAM (utente o ruolo) di aggiungere, elencare o rimuovere un tag per un dispositivo MFA virtuale

Aggiungi l'istruzione seguente alla policy di autorizzazione per l'entità IAM che deve gestire i tag. Utilizza il tuo numero di account e sostituisci *`<MFATokenID>`* con il nome del dispositivo MFA virtuale di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListMFADeviceTags",
```

```
        "iam:TagMFADevice",
        "iam:UntagMFADevice"
    ],
    "Resource": "arn:aws:iam::<account-number>:mfa/<MFATokenID>"
}
```

Come consentire a un'entità IAM (utente o ruolo) di aggiungere un tag a un determinato dispositivo MFA virtuale

Aggiungi l'istruzione seguente alla policy di autorizzazione dell'entità IAM che potrà aggiungere ma non rimuovere i tag per uno specifico dispositivo MFA virtuale.

> **Note**
>
> L'azione `iam:TagMFADevice` richiede che tu includa anche l'azione `iam:ListMFADeviceTags`.

Per utilizzare questa policy, sostituisci *<MFATokenID>* con il nome del dispositivo MFA virtuale di cui è necessario gestire i tag. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:ListMFADeviceTags",
        "iam:TagMFADevice"
    ],
    "Resource": "arn:aws:iam::<account-number>:mfa/<MFATokenID>"
}
```

In alternativa, puoi utilizzare una policy AWS gestita come IAM FullAccess per fornire l'accesso completo a IAM.

Gestione dei tag su dispositivi MFA virtuali (AWS CLI o AWS API)

È possibile elencare, allegare o rimuovere i tag di un dispositivo MFA virtuale. È possibile utilizzare l'API AWS CLI o l' AWS API per gestire i tag per un dispositivo MFA virtuale.

Per elencare i tag attualmente collegati a un dispositivo MFA virtuale (AWS CLI o AWS API)

- AWS CLI: era iam list-mfa-device-tags
- AWS API: ListMFA DeviceTags

Per allegare tag a un dispositivo MFA virtuale (AWS CLI o AWS API)

- AWS CLI: è stato tag-mfa-device
- AWS API: TagMFADevice

Per rimuovere i tag da un dispositivo MFA virtuale (AWS CLI o AWS API)

- AWS CLI: è stato untag-mfa-device
- AWS API: unTagMFADevice

Per informazioni su come allegare tag alle risorse per altri AWS servizi, consulta la documentazione relativa a tali servizi.

Per ulteriori informazioni sull'utilizzo di tag per impostare autorizzazioni più granulari con le policy di autorizzazione IAM, consulta Elementi delle policy IAM: variabili e tag.

Passare i tag di sessione AWS STS

I tag di sessione sono attributi di coppia chiave-valore che vengono passati quando si assume un ruolo IAM o si federa un utente in AWS STS. Puoi farlo effettuando una richiesta AWS CLI o AWS API tramite AWS STS o tramite il tuo provider di identità (IdP). Quando richiedi credenziali di sicurezza temporanee, generi una sessione. AWS STS Le sessioni scadono e dispongono di credenziali, ad esempio una coppia di chiavi di accesso e un token di sessione. Quando si utilizzano le credenziali di sessione per effettuare una richiesta successiva, il contesto della richiesta include la chiave di contesto `aws:PrincipalTag`. È possibile utilizzare la chiave `aws:PrincipalTag` nell'elemento `Condition` delle proprie policy per consentire o negare l'accesso in base a tali tag.

Quando si utilizzano credenziali temporanee per effettuare una richiesta, l'entità potrebbe includere un set di tag. Questi tag provengono dalle seguenti fonti:

1. Tag di sessione: i tag passati quando assumi il ruolo o federi l'utente utilizzando l'API AWS CLI o AWS . Per ulteriori informazioni su queste operazioni, consulta Operazioni di tagging di sessione.

2. Tag di sessione transitivi in ingresso: questi tag sono stati ereditati da una sessione precedente in un concatenamento di ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione più avanti in questo argomento.
3. Tag IAM: i tag associati al ruolo IAM assunto.

Argomenti

Operazioni di tagging di sessione

Puoi passare i tag di sessione utilizzando quanto segue AWS CLI o le operazioni AWS API in AWS STS. La funzione AWS Management Console Switch Role non consente di passare i tag di sessione.

È inoltre possibile impostare i tag di sessione come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Confronto dei metodi per il passaggio dei tag di sessione

Operazione	Chi può assumere il ruolo	Metodo di passaggio dei tag	Metodo di impostazione dei tag transitivi
`assume-role` CLI oppure	Utente IAM o sessione	Parametro API `Tags` o opzione CLI `--tags`	Parametro API `TransitiveTagKeys` o

Operazione	Chi può assumere il ruolo	Metodo di passaggio dei tag	Metodo di impostazione dei tag transitivi
operazione API `AssumeRole`			opzione CLI `--transitive-tag-keys`
`assume-role-with-saml` CLI oppure operazione API `AssumeRoleWithSAML`	Tutti gli utenti autenticati che utilizzano un provider di identità SAML	Attributo SAML `PrincipalTag`	Attributo SAML `TransitiveTagKeys`
`assume-role-with-web-identity` CLI oppure operazione API `AssumeRoleWithWebIdentity`	Qualsiasi utente autenticato utilizzando un provider OIDC	`PrincipalTag` Token OIDC	`TransitiveTagKeys` Token OIDC
`get-federation-token` CLI oppure operazione API `GetFederationToken`	Utente IAM o utente root	Parametro API `Tags` o opzione CLI `--tags`	Non supportato

Le operazioni che supportano i tag di sessione potrebbero non concludersi correttamente se si verifica una delle seguenti condizioni:

- Sono passati oltre 50 tag di sessione.
- Il testo in chiaro delle chiavi dei tag di sessione supera i 128 caratteri.
- Il testo in chiaro dei valori dei tag di sessione supera i 256 caratteri.
- La dimensione totale del testo in chiaro delle policy di sessione supera i 2048 caratteri.

- La dimensione totale del pacchetto delle policy e dei tag di sessione combinati è troppo grande. Se l'operazione non ha esito positivo, il messaggio di errore mostra quanto le policy e i tag combinati si avvicinano al limite di dimensione superiore, in percentuale.

Cose da sapere sui tag di sessione

Prima di utilizzare i tag di sessione, esaminare i seguenti dettagli sulle sessioni e sui tag.

- Quando utilizzi i tag di sessione, le policy di attendibilità per tutti i ruoli connessi al provider di identità (IdP) che passa i tag devono disporre dell'autorizzazione `sts:TagSession`. Per i ruoli che non dispongono di questa autorizzazione nella policy di attendibilità, l'operazione `AssumeRole` avrà esito negativo.
- Quando si richiede una sessione, è possibile specificare i tag principali come tag di sessione. I tag si applicano alle richieste effettuate utilizzando le credenziali della sessione.
- I tag di sessione sono coppie chiave-valore. Ad esempio, per aggiungere informazioni di contatto a una sessione, è possibile aggiungere la chiave del tag di sessione `email` e il valore del tag `johndoe@example.com`.
- I tag di sessione devono seguire le regole per la denominazione dei tag in IAM e. AWS STS In questo argomento sono incluse informazioni sulla distinzione tra maiuscole e minuscole e sui prefissi riservati validi per i tag di sessione.
- I nuovi tag di sessione sovrascrivono i tag relativi ai ruoli assunti o agli utenti federati con la stessa chiave di tag, indipendentemente dall'utilizzo di lettere minuscole o maiuscole.
- Non è possibile passare i tag di sessione utilizzando. AWS Management Console
- I tag di sessione sono validi solo per la sessione corrente.
- I tag di sessione supportano il concatenamento dei ruoli. Per impostazione predefinita, AWS STS non passa tag alle sessioni di ruolo successive. Tuttavia, è possibile impostare i tag di sessione come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli e sostituiscono i valori `ResourceTag` corrispondenti dopo la valutazione della policy di attendibilità del ruolo. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.
- È possibile utilizzare i tag di sessione per controllare l'accesso alle risorse o per controllare quali tag possono essere passati in una sessione successiva. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- È possibile visualizzare i tag del principale per la sessione, inclusi i tag di sessione, nei log AWS CloudTrail . Per ulteriori informazioni, consulta Visualizzazione dei tag di sessione in CloudTrail.

- È necessario passare un singolo valore per ogni tag di sessione. AWS STS non supporta tag di sessione multivalore.
- È possibile passare un massimo di 50 tag di sessione. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.
- Una AWS conversione comprime le politiche di sessione e i tag di sessione passati combinati in un formato binario compresso con un limite separato. Se superi questo limite, il messaggio di errore AWS CLI o AWS API mostra quanto le politiche e i tag combinati si avvicinano al limite di dimensione superiore, in percentuale.

Autorizzazioni necessarie per aggiungere tag di sessione

Oltre a quella sull'operazione che corrisponde all'operazione API, è necessario disporre nella policy dell'autorizzazione per le seguenti operazioni:

```
sts:TagSession
```

> **Important**
>
> Quando si utilizzano i tag di sessione, i criteri di attendibilità dei ruoli per tutti i ruoli connessi a un provider di identità (IdP) devono disporre dell'autorizzazione `sts:TagSession`. L'operazione `AssumeRole` avrà esito negativo per qualsiasi ruolo connesso a un provider di identità che passa tag di sessione senza questa autorizzazione. Se non si desidera aggiornare la policy di attendibilità del ruolo per ogni ruolo, è possibile utilizzare un'istanza IdP separata per passare i tag di sessione. Quindi, aggiungi l'autorizzazione `sts:TagSession` solo ai ruoli connessi all'IdP separato.

È possibile utilizzare l'operazione `sts:TagSession` con le seguenti chiavi di condizione.

- `aws:PrincipalTag`: confronta il tag collegato al principale che effettua la richiesta con il tag specificato nella policy. Ad esempio, è possibile consentire a un'entità di passare i tag di sessione solo se l'entità che effettua la richiesta dispone dei tag specificati.
- `aws:RequestTag`: confronta la coppia chiave-valore del tag passata nella richiesta con la coppia del tag specificata nella policy. Ad esempio, è possibile consentire all'entità di passare tag di sessione specificati, ma solo con i valori specificati.

- `aws:ResourceTag`: confronta la coppia chiave-valore tag specificata nella policy con la coppia chiave-valore collegata alla risorsa. Ad esempio, puoi consentire al principale di passare i tag di sessione solo se il ruolo che assume include i tag specificati.
- `aws:TagKeys`: confronta le chiavi tag in una richiesta con quelle specificate nella policy. Ad esempio, è possibile consentire all'entità di passare solo i tag di sessione con le chiavi dei tag specificate. Questa chiave di condizione limita l'insieme massimo di tag di sessione che possono essere passati.
- `sts:TransitiveTagKeys`: confronta le chiavi dei tag di sessione transitivi nella richiesta con quelle specificate nella policy. Ad esempio, è possibile scrivere una policy per consentire a un'entità di impostare solo tag specifici come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Ad esempio, la seguente policy di attendibilità del ruolo consente all'utente `test-session-tags` di assumere il ruolo a cui è collegata la policy. Quando l'utente assume il ruolo, deve utilizzare l' AWS API AWS CLI o per passare i tre tag di sessione richiesti e l'ID esterno richiesto. Inoltre, l'utente può scegliere di impostare i tag `Department` e `Project` come transitivi.

Example Esempio di policy di attendibilità dei ruoli per i tag di sessione

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIamUserAssumeRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::123456789012:user/test-session-tags"},
            "Condition": {
              "StringLike": {
                    "aws:RequestTag/Project": "*",
                    "aws:RequestTag/CostCenter": "*",
                    "aws:RequestTag/Department": "*"
                },
                "StringEquals": {"sts:ExternalId": "Example987"}
            }
        },
        {
            "Sid": "AllowPassSessionTagsAndTransitive",
            "Effect": "Allow",
```

```
            "Action": "sts:TagSession",
            "Principal": {"AWS": "arn:aws:iam::123456789012:user/test-session-tags"},
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/Project": "*",
                    "aws:RequestTag/CostCenter": "*"
                },
                "StringEquals": {
                    "aws:RequestTag/Department": [
                        "Engineering",
                        "Marketing"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "sts:TransitiveTagKeys": [
                        "Project",
                        "Department"
                    ]
                }
            }
        }
    ]
}
```

Che cosa fa questa policy?

- L'istruzione `AllowIamUserAssumeRole` consente all'utente `test-session-tags` di assumere il ruolo a cui è collegata la policy. Quando tale utente assume il ruolo, deve passare i tag di sessione richiesti e l'ID esterno.
  - Il primo blocco condizionale di questa istruzione richiede all'utente di passare i tag di sessione `Project`, `CostCenter` e `Department`. I valori dei tag non sono significativi in questa istruzione, quindi abbiamo usato caratteri jolly (*) per i valori dei tag. Questo blocco assicura che l'utente passi almeno questi tre tag di sessione. In caso contrario, l'operazione non va a buon fine. L'utente può passare tag aggiuntivi.
  - Il secondo blocco condizionale richiede all'utente di passare un ID esterno con il valore `Example987`.
- L'istruzione `AllowPassSessionTagsAndTransitive` autorizza l'operazione `sts:TagSession`. Questa operazione deve essere autorizzata prima che l'utente possa passare i tag di sessione. Se la policy include la prima istruzione senza la seconda, l'utente non può assumere il ruolo.

- Il primo blocco condizionale di questa istruzione consente all'utente di passare qualsiasi valore per i tag di sessione `CostCenter` e `Project`. A tale scopo, è necessario utilizzare i caratteri jolly (*) per il valore del tag nella policy, il che richiede l'utilizzo dell'operatore di StringLikecondizione.
- Il secondo blocco condizionale consente all'utente di passare solo i valori `Marketing` o `Engineering` per il tag di sessione `Department`.
- Il terzo blocco condizionale elenca l'insieme massimo di tag che possono essere impostati come transitivi. L'utente può scegliere di impostare un sottoinsieme o nessun tag come transitivo. Ma non può impostare tag aggiuntivi come transitivi. È possibile richiedere che imposti almeno uno dei tag come transitivo aggiungendo un altro blocco condizionale che include `"Null": {"sts:TransitiveTagKeys":"false"}`.

Passare i tag di sessione utilizzando AssumeRole

L'`AssumeRole`operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. È possibile utilizzare le credenziali dell'utente o del ruolo IAM per chiamare `AssumeRole`. Per passare i tag di sessione mentre assumi un ruolo, utilizzate l'`--tags` AWS CLI opzione o il parametro `Tags` AWS API.

Per impostare i tag come transitivi, utilizzate l'`--transitive-tag-keys` AWS CLI opzione o il parametro `TransitiveTagKeys` AWS API. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Nell'esempio seguente viene illustrata una richiesta di esempio che utilizza `AssumeRole`. In questo esempio, quando si assume il ruolo `my-role-example`, si crea una sessione denominata `my-session`. Aggiungere le coppie chiave-valore dei tag di sessione `Project` = `Automation`, `CostCenter` = `12345` e `Department` = `Engineering`. È inoltre possibile impostare i tag `Project` e `Department` come transitivi specificando le loro chiavi.

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/my-role-example \
--role-session-name my-session \
--tags Key=Project,Value=Automation Key=CostCenter,Value=12345
 Key=Department,Value=Engineering \
--transitive-tag-keys Project Department \
--external-id Example987
```

Passaggio dei tag di sessione tramite SAML AssumeRoleWith

L'operazione `AssumeRoleWithSAML` viene autenticata con la federazione basata su SAML. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere AWS alle risorse. Per ulteriori informazioni sull'utilizzo della federazione basata su SAML per AWS Management Console l'accesso, consulta. Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console Per dettagli sull'accesso alle AWS CLI nostre AWS API, consulta. Federazione SAML 2.0 Per un tutorial sulla configurazione della federazione SAML per gli utenti di Active Directory, consulta AWS Federated Authentication with Active Directory Federation Services (ADFS) nel Security Blog. AWS

In qualità di amministratore, puoi consentire ai membri della tua directory aziendale di unirsi per utilizzare l'operazione. AWS AWS STS `AssumeRoleWithSAML` A tale scopo, è necessario completare le seguenti attività:

1. Configurazione della rete come un provider SAML per AWS
2. Creazione di un provider SAML in IAM
3. Configurazione di un ruolo e delle autorizzazioni in AWS per gli utenti federati
4. Termine della configurazione del provider di identità SAML e creazione di asserzioni per la risposta di autenticazione SAML

AWS include fornitori di identità con end-to-end esperienza certificata per i tag di sessione con le loro soluzioni di identità. Per informazioni su come utilizzare questi provider di identità per configurare i tag di sessione, consultare Integra fornitori di soluzioni SAML di terze parti con AWS.

Per passare gli attributi SAML come tag di sessione, includere l'elemento `Attribute` con l'attributo `Name` impostato a `https://aws.amazon.com/SAML/Attributes/PrincipalTag:`*`{TagKey}`*. Utilizzare l'elemento `AttributeValue` per specificare il valore del tag. Includere un elemento `Attribute` separato per ogni tag di sessione.

Ad esempio, si supponga di voler passare i seguenti attributi di identità come tag di sessione:

- `Project:Automation`
- `CostCenter:12345`
- `Department:Engineering`

Per passare questi attributi, includere i seguenti elementi nell'asserzione SAML.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
  <AttributeValue>Automation</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:CostCenter">
  <AttributeValue>12345</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Department">
  <AttributeValue>Engineering</AttributeValue>
</Attribute>
```

Per impostare i tag sopra elencati come transitivi, include un altro elemento `Attribute` con l'attributo Name impostato su `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Per impostare i tag `Project` e `Department` come transitivi, utilizzare il seguente attributo multi valore.

Example Esempio di frammento di un'asserzione SAML

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>Project</AttributeValue>
  <AttributeValue>Department</AttributeValue>
</Attribute>
```

Passare i tag di sessione utilizzando AssumeRoleWithWebIdentity

Usa la federazione conforme a OpenID Connect (OIDC) per autenticare l'operazione. `AssumeRoleWithWebIdentity` Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle risorse. AWS Per ulteriori informazioni sull'utilizzo della federazione delle identità Web per AWS Management Console l'accesso, vedereFederazione OIDC.

Per passare i tag di sessione da OpenID Connect (OIDC), è necessario includere i tag di sessione nel Token Web JSON (JWT). Includere i tag di sessione nello spazio dei nomi `https://aws.amazon.com/` `tags` quando si invia la richiesta `AssumeRoleWithWebIdentity`. Per ulteriori informazioni sui token e le registrazioni OIDC, consultare Utilizzo di token con pool di utenti nella Guida per gli sviluppatori Amazon Cognito .

Ad esempio, il seguente JWT decodificato utilizza un token per chiamare `AssumeRoleWithWebIdentity` con i tag di sessione `Project`, `CostCenter` e `Department`. Il token imposta anche i tag `Project` e `CostCenter` come transitivi. I tag transitivi persistono durante il concatenamento dei ruoli. Per ulteriori informazioni, consulta Concatenamento dei ruoli con i tag di sessione.

Example Esempio di token Web JSON decodificato

```
{
    "sub": "johndoe",
    "aud": "ac_oic_client",
    "jti": "ZYUCeRMQVtqHypVPWAN3VB",
    "iss": "https://xyz.com",
    "iat": 1566583294,
    "exp": 1566583354,
    "auth_time": 1566583292,
    "https://aws.amazon.com/tags": {
        "principal_tags": {
            "Project": ["Automation"],
            "CostCenter": ["987654"],
            "Department": ["Engineering"]
        },
        "transitive_tag_keys": [
            "Project",
            "CostCenter"
        ]
    }
}
```

Passare i tag di sessione utilizzando GetFederationToken

La `GetFederationToken` consente di federare l'utente. Questa operazione restituisce un set di credenziali temporanee che è possibile utilizzare per accedere alle AWS risorse. Per aggiungere tag alla sessione utente federata, utilizzate l'`--tags` AWS CLI opzione o il parametro `Tags` AWS API. Non è possibile impostare i tag di sessione come transitivi quando si utilizza `GetFederationToken`, perché non è possibile utilizzare le credenziali provvisorie per assumere un ruolo. In questo caso non è possibile utilizzare il concatenamento dei ruoli.

Di seguito è mostrata una risposta di esempio che utilizza `GetFederationToken`. In questo esempio, quando si richiede il token, si crea una sessione denominata `my-fed-user`. Aggiungere le coppie chiave-valore dei tag di sessione `Project` = `Automation` e `Department` = `Engineering`.

Example Richiesta GetFederationToken CLI di esempio

```
aws sts get-federation-token \
--name my-fed-user \
--tags key=Project,value=Automation key=Department,value=Engineering
```

Quando si utilizzano le credenziali temporanee restituite dall'operazione `GetFederationToken`, i tag del principale della sessione includono i tag dell'utente e i tag di sessione passati.

Concatenamento dei ruoli con i tag di sessione

È possibile assumere un ruolo e quindi utilizzare le credenziali temporanee per assumere un altro ruolo. È possibile continuare da una sessione all'altra. Questa operazione è chiamata concatenamento del ruolo. Quando si passano i tag di sessione mentre si assume un ruolo, è possibile impostare le chiavi come transitive. Ciò garantisce che tali tag di sessione siano passati alle sessioni successive in un concatenamento di ruoli. Non è possibile impostare tag di ruolo come transitivi. Per passare questi tag alle sessioni successive, specificarli come tag di sessione.

> **Note**
>
> I tag transitivi persistono durante il concatenamento dei ruoli e sostituiscono i valori `ResourceTag` corrispondenti dopo la valutazione della policy di attendibilità del ruolo.

L'esempio seguente mostra come AWS STS passa i tag di sessione, i tag transitivi e i tag di ruolo nelle sessioni successive in una catena di ruoli.

In questo esempio di scenario di concatenamento dei ruoli, si utilizza una chiave di accesso utente IAM AWS CLI per assumere un ruolo denominato `Role1` È quindi possibile utilizzare le credenziali di sessione risultanti per assumere un secondo ruolo denominato `Role2`. È quindi possibile utilizzare le credenziali della seconda sessione per assumere un terzo ruolo denominato `Role3`. Queste richieste sono eseguite come tre operazioni separate. Ogni ruolo è già taggato in IAM. E durante ogni richiesta, è possibile passare ulteriori tag di sessione.

Quando si concatenano i ruoli, è possibile assicurarsi che i tag di una sessione precedente persistano nelle sessioni successive. Per fare ciò utilizzando il comando `assume-role` della CLI, è necessario passare il tag come tag di sessione e impostare il tag come transitivo. Si passa il tag `Star = 1` come tag di sessione. Il comando collega anche il tag `Heart = 1` al ruolo e lo applica come tag del principale quando si utilizza la sessione. Tuttavia, si desidera anche che il tag `Heart = 1`

sia passato automaticamente alla seconda o terza sessione. Per farlo, è necessario includerlo manualmente come tag di sessione. I tag principali di sessione risultanti includono entrambi questi tag e li impostano come transitivi.

Esegui questa richiesta utilizzando il seguente AWS CLI comando:

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role1 \
--role-session-name Session1 \
--tags Key=Star,Value=1 Key=Heart,Value=1 \
--transitive-tag-keys Star Heart
```

È quindi possibile utilizzare le credenziali di tale sessione per assumere il `Role2`. Il comando collega il tag `Sun = 2` al secondo ruolo e lo applica come tag del principale quando utilizzi la sessione. I tag `Heart` e `Star` ereditano dai tag di sessione transitivi nella prima sessione. I tag del principale della seconda sessione risultanti sono `Heart = 1`, `Star = 1` e `Sun = 2`. `Heart` e `Star` continueranno a essere transitivi. Il tag `Sun` collegato a `Role2` non è contrassegnato come transitivo perché non è un tag di sessione. Le sessioni future non ereditano questo tag.

Esegui questa seconda richiesta utilizzando il seguente AWS CLI comando:

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role2 \
--role-session-name Session2
```

È quindi possibile utilizzare le credenziali della seconda sessione per assumere il `Role3`. I tag dell'entità per la terza sessione derivano da tutti i nuovi tag di sessione, i tag di sessione transitivi ereditati e i tag di ruolo. I tag `Heart = 1` e `Star = 1` nella seconda sessione sono stati ereditati dai tag di sessione transitivi nella prima sessione. Se provi a passare il tag di sessione `Sun = 2`, l'operazione avrà esito negativo. Il tag di sessione `Star = 1` ereditato sostituisce il tag `Star = 3` del ruolo. Nella concatenazione dei ruoli, il valore di un tag transitivo sovrascrive il ruolo che corrisponde al valore `ResourceTag` dopo della valutazione della policy di attendibilità del ruolo. In questo esempio, se `Role3` utilizza `Star` come `ResourceTag` nella policy di attendibilità ruolo e imposta il valore `ResourceTag` sul valore del tag transitivo dalla sessione del ruolo chiamante. Il tag del ruolo `Lightning` si applica anche alla terza sessione e non è impostato come transitivo.

La terza richiesta viene eseguita utilizzando il seguente AWS CLI comando:

Example Richiesta AssumeRole CLI di esempio

```
aws sts assume-role \
--role-arn arn:aws:iam::123456789012:role/Role3 \
--role-session-name Session3
```

Utilizzo dei tag di sessione per ABAC

Il controllo degli accessi basato su attributi (Attribute-Based Access Control, ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi dei tag.

Se l'azienda utilizza un provider di identità (IdP) basato su SAML o OIDC per gestire le identità utente, puoi configurare l'asserzione per passare i tag di sessione ad AWS. Ad esempio, con le identità utente aziendali, quando i dipendenti si uniscono AWS, AWS applica i loro attributi al principale risultante. È quindi possibile utilizzare ABAC per consentire o negare le autorizzazioni sulla base di tali attributi. Per informazioni dettagliate, vedi Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.

Per ulteriori informazioni sull'utilizzo di IAM Identity Center con ABAC, consulta Attributi per il controllo degli accessi nella Guida per l'utente di AWS IAM Identity Center .

Visualizzazione dei tag di sessione in CloudTrail

È possibile utilizzare AWS CloudTrail per visualizzare le richieste utilizzate per assumere ruoli o federare gli utenti. Il file di CloudTrail registro include informazioni sui tag principali per la sessione utente con ruolo assunto o federata. Per ulteriori informazioni, consulta Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail.

Ad esempio, supponiamo di effettuare una AWS STS `AssumeRoleWithSAML` richiesta, passare i tag di sessione e impostare tali tag come transitivi. Puoi trovare le seguenti informazioni nel tuo CloudTrail registro.

Example Esempio di AssumeRoleWith registro SAML CloudTrail

```
    "requestParameters": {
        "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
        "roleSessionName": "MyRoleSessionName",
```

```
        "principalTags": {
            "CostCenter": "987654",
            "Project": "Unicorn"
        },
        "transitiveTagKeys": [
            "CostCenter",
            "Project"
        ],
        "durationSeconds": 3600,
        "roleArn": "arn:aws:iam::123456789012:role/SAMLTestRoleShibboleth",
        "principalArn": "arn:aws:iam::123456789012:saml-provider/Shibboleth"
    },
```

È possibile visualizzare i seguenti CloudTrail registri di esempio per visualizzare gli eventi che utilizzano tag di sessione.

- Esempio di evento API di concatenamento dei AWS STS ruoli nel file di registro CloudTrail
- Esempio di evento AWS STS API SAML nel file di registro CloudTrail
- Esempio di evento AWS STS API OIDC nel CloudTrail file di registro

Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail

IAM e AWS STS sono integrati con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente o un ruolo IAM. CloudTrail acquisisce tutte le chiamate API per IAM e AWS STS come eventi, incluse le chiamate dalla console e dalle chiamate API. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Puoi utilizzarlo CloudTrail per ottenere informazioni sulla richiesta che è stata fatta a IAM o AWS STS. Ad esempio, puoi visualizzare l'indirizzo IP da cui è stata Effettuata la richiesta, l'autore della richiesta e il momento in cui è stata effettuata, nonché dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

Argomenti

- IAM e AWS STS informazioni in CloudTrail
- Registrazione delle richieste IAM e API AWS STS
- Registrazione di richieste API ad altri servizi AWS

- Registrazione di eventi di accesso dell'utente
- Registrazione di eventi di accesso per credenziali temporanee
- Esempi di eventi dell'API IAM nel CloudTrail registro
- Esempi di eventi AWS STS API nel registro CloudTrail
- Esempio di eventi di accesso nel log CloudTrail
- IAM role trust policy, comportamento

IAM e AWS STS informazioni in CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in IAM or AWS STS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti in Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi del tuo Account AWS, inclusi gli eventi per IAM e AWS STS, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni . Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutti gli IAM e AWS STS le azioni vengono registrati CloudTrail e documentati negli IAM API Reference e API Reference.AWS Security Token Service

Registrazione delle richieste IAM e API AWS STS

CloudTrail registra tutte le richieste API autenticate su IAM e AWS STS sulle operazioni API. CloudTrail registra inoltre le richieste non autenticate alle AWS STS azioni `AssumeRoleWithSAML` e `AssumeRoleWithWebIdentity` registra le informazioni fornite dal provider di identità. Tuttavia,

alcune AWS STS richieste non autenticate potrebbero non essere registrate perché non soddisfano l'aspettativa minima di essere sufficientemente valide da essere considerate una richiesta legittima.

È possibile utilizzare le informazioni registrate per mappare le chiamate effettuate da un utente federato con un ruolo assunto al chiamante federato esterno di origine. Nel caso di`AssumeRole`, è possibile mappare le chiamate al AWS servizio di origine o all'account dell'utente di origine. La `userIdentity` sezione dei dati JSON nella voce di CloudTrail registro contiene le informazioni necessarie per mappare la AssumeRole* richiesta con un utente federato specifico. Per ulteriori informazioni, consulta CloudTrail UserIdentity Element nella Guida per l'AWS CloudTrail utente.

Ad esempio, le chiamate a IAM `CreateUser` e `DeleteRole` `ListGroups` ad altre operazioni API vengono tutte registrate da. CloudTrail

Esempi di questo tipo di voce di log sono riportati più avanti in questo argomento.

Registrazione di richieste API ad altri servizi AWS

Le richieste autenticate ad altre operazioni dell'API di AWS servizio vengono registrate da CloudTrail e queste voci di registro contengono informazioni su chi ha generato la richiesta.

Ad esempio, supponiamo di effettuare una richiesta per elencare le istanze Amazon EC2 o creare un gruppo di implementazione AWS CodeDeploy . I dettagli sulla persona o sul servizio che ha effettuato la richiesta sono contenuti nella voce di log per quella richiesta. Queste informazioni consentono di determinare se la richiesta è stata effettuata da un utente IAM Utente root dell'account AWS, da un ruolo o da un altro AWS servizio.

Per ulteriori dettagli sulle informazioni sull'identità dell'utente nelle voci di CloudTrail registro, vedere UserIdentity Element nella Guida per l'AWS CloudTrail utente.

Registrazione di eventi di accesso dell'utente

CloudTrail registra gli eventi di accesso ai AWS Management Console, ai forum di AWS discussione e. Marketplace AWS CloudTrailregistra i tentativi di accesso riusciti e non riusciti per gli utenti IAM e gli utenti federati.

Per visualizzare CloudTrail gli eventi di esempio relativi agli accessi riusciti e non riusciti degli utenti root, consulta Example event records for root users nella User Guide.AWS CloudTrail

Come procedura consigliata in materia di sicurezza, AWS non registra il testo del nome utente IAM immesso quando l'errore di accesso è causato da un nome utente errato. Il nome utente viene

mascherato dal valore `HIDDEN_DUE_TO_SECURITY_REASONS`. Per un esempio di questo caso, consultare Esempio di evento di accesso non riuscito a causa di un nome utente non corretto più avanti in questo argomento. Il testo del nome utente viene oscurato perché tali errori potrebbero essere causati da errori utente. La registrazione di questi errori potrebbe esporre informazioni potenzialmente sensibili. Ad esempio:

- L'utente ha immesso accidentalmente la password nella casella del nome utente.
- Scegli il link per la pagina di accesso di uno Account AWS, ma poi digita il numero di account per un'altra. Account AWS
- L'utente ha dimenticato a quale account stava eseguendo l'accesso e ha accidentalmente digitato il nome dell'account e-mail personale, l'identificatore di accesso bancario o un altro ID privato.

Registrazione di eventi di accesso per credenziali temporanee

Quando un principale richiede credenziali temporanee, il tipo principale determina la modalità di CloudTrail registrazione dell'evento. Questo può essere complicato quando un principale assume un ruolo in un altro account. Esistono più chiamate API per eseguire operazioni correlate a operazioni multiaccount al ruolo. Innanzitutto, il principale chiama un' AWS STS API per recuperare le credenziali temporanee. Tale operazione viene registrata nell'account chiamante e nell'account in cui viene AWS STS eseguita l'operazione. Quindi l'entità principale utilizza il ruolo per eseguire altre chiamate API nell'account del ruolo assunto.

È possibile utilizzare la chiave di condizione `sts:SourceIdentity` nella policy di attendibilità del ruolo per richiedere agli utenti di specificare un'identità quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come identità di origine. In questo modo è possibile determinare quale utente ha eseguito un'operazione specifica in AWS. Per ulteriori informazioni, consulta `sts:SourceIdentity`. Puoi utilizzare `sts:RoleSessionName` anche per richiedere agli utenti di specificare un nome di sessione quando assumono un ruolo. Questo può aiutarti a distinguere tra le sessioni di ruolo per un ruolo utilizzato da diversi responsabili durante la revisione dei registri. AWS CloudTrail

La tabella seguente mostra come CloudTrail registra le diverse informazioni sull'identità utente per ciascuna delle AWS STS API che generano credenziali temporanee.

Tipo di entità principale	API STS	Identità utente nel CloudTrail registro dell'account del chiamante	Identità dell'utente nel CloudTrail registro dell'account del ruolo assunto	Identità dell'utente nel CloudTrail registro per le successive chiamate API del ruolo
Utente root dell'account AWS credenziali	GetSessionToken	Identità utente root	L'account proprietario del ruolo è uguale all'account chiamante	Identità utente root
Utente IAM	GetSessionToken	Identità utente IAM	L'account proprietario del ruolo è uguale all'account chiamante	Identità utente IAM
Utente IAM	GetFederationToken	Identità utente IAM	L'account proprietario del ruolo è uguale all'account chiamante	Identità utente IAM
Utente IAM	AssumeRole	Identità utente IAM	numero di conto e ID principale (se si tratta di un utente) o principale AWS del servizio	Solo identità ruolo (nessun utente)
Utente autenticato esternamente	AssumeRoleWithSAML	N/A	Identità utente SAML	Solo identità ruolo (nessun utente)

Tipo di entità principale	API STS	Identità utente nel CloudTrail registro dell'account del chiamante	Identità dell'utente nel CloudTrail registro dell'account del ruolo assunto	Identità dell'utente nel CloudTrail registro per le successive chiamate API del ruolo
Utente autenticato esternamente	AssumeRoleWithWebIdentity	N/A	Identità utente OIDC/Web	Solo identità ruolo (nessun utente)

CloudTrail considera un'azione di sola lettura se non ha alcun effetto mutante su una risorsa. Quando si registra un evento di sola CloudTrail lettura, oscura le informazioni nel registro. `responseElements` Quando CloudTrail registra un evento che non è di sola lettura, il dato completo `responseElements` viene visualizzato nella voce di registro. Tuttavia, per le AWS STS API`AssumeRole`, e `AssumeRoleWithSAML``AssumeRoleWithWebIdentity`, anche se sono registrate come di sola lettura, CloudTrail includerà il dato completo `responseElements` nel registro di queste API.

La tabella seguente mostra i CloudTrail log `responseElements` e le `readOnly` informazioni per ciascuna delle API che generano credenziali temporanee. AWS STS

API STS	Informazioni sugli elementi di risposta	Sola lettura
AssumeRole	Incluse	true
AssumeRoleWithSAML	Incluso	true
AssumeRoleWithWebIdentity	Incluso	true
GetFederationToken	Incluso	false
GetSessionToken	Incluso	false

Esempi di eventi dell'API IAM nel CloudTrail registro

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento API rappresenta una singola richiesta API e include informazioni sul principale, l'operazione richiesta, gli eventuali parametri e la data e l'ora dell'operazione.

Esempio di evento API IAM nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l'`GetUserPolicy`azione IAM.


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/JaneDoe",
    "accountId": "444455556666",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "JaneDoe",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2014-07-15T21:39:40Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2014-07-15T21:40:14Z",
  "eventSource": "iam.amazonaws.com",
  "eventName": "GetUserPolicy",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "signin.amazonaws.com",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "userName": "JaneDoe",
    "policyName": "ReadOnlyAccess-JaneDoe-201407151307"
  },
  "responseElements": null,
  "requestID": "9EXAMPLE-0c68-11e4-a24e-d5e16EXAMPLE",
  "eventID": "cEXAMPLE-127e-4632-980d-505a4EXAMPLE"
}
```

Da queste informazioni sull'evento puoi determinare che la richiesta è stata effettuata per ottenere una policy utente denominata `ReadOnlyAccess-JaneDoe-201407151307` per l'utente `JaneDoe`, come specificato nell'elemento `requestParameters`. Puoi anche consultare che la richiesta è stata effettuata da un utente IAM denominato `JaneDoe` in data 15 luglio 2014 alle 21:40 UTC. In questo caso, la richiesta ha avuto origine in AWS Management Console, come si può vedere dall'`userAgent`elemento.

Esempi di eventi AWS STS API nel registro CloudTrail

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento API rappresenta una singola richiesta API e include informazioni sul principale, l'operazione richiesta, gli eventuali parametri e la data e l'ora dell'operazione.

Esempi di eventi AWS STS API tra account nei file di registro CloudTrail

L'utente IAM indicato `JohnDoe` nell'account 777788889999 richiama l' AWS STS `AssumeRole`azione per assumere il ruolo `EC2-dev` nell'account 111122223333. Quando si assume il ruolo, l'amministratore dell'account richiede agli utenti di impostare un'identità di origine uguale al proprio nome utente. L'utente passa il valore dell'identità di origine di `JohnDoe`.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "arn": "arn:aws:iam::777788889999:user/JohnDoe",
    "accountId": "777788889999",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "JohnDoe"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "JohnDoe-EC2-dev",
    "sourceIdentity": "JohnDoe",
```

```
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
      "expiration": "Jul 18, 2023, 4:07:39 PM"
      },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:JohnDoe-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/JohnDoe-EC2-dev"
    },
  "sourceIdentity": "JohnDoe"
  },
  "resources": [
    {
      "ARN": "arn:aws:iam::111122223333:role/EC2-dev",
      "accountId": "111122223333",
      "type": "AWS::IAM::Role"
    }
  ],
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Il secondo esempio mostra la voce di CloudTrail registro dell'account del ruolo assunto (111122223333) per la stessa richiesta.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AWSAccount",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "accountId": "777788889999"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
```

```
  "userAgent": "aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "JohnDoe-EC2-dev",
    "sourceIdentity": "JohnDoe",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
      "expiration": "Jul 18, 2014, 4:07:39 PM"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:JohnDoe-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/JohnDoe-EC2-dev"
      },
  "sourceIdentity": "JohnDoe"
  },
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE"
}
```

Esempio di evento API di concatenamento dei AWS STS ruoli nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata da John Doe nell'account 1111. John in precedenza utilizzava il suo utente `JohnDoe` per assumere il ruolo `JohnRole1`. Per questa richiesta, utilizza le credenziali di tale ruolo per assumere il ruolo `JohnRole2`. Questo è noto come concatenazione del ruolo. L'identità di origine che ha impostato quando ha assunto il ruolo `JohnDoe1` persiste nella richiesta per assumere `JohnRole2`. Se John prova a impostare un'identità di origine diversa quando assume il ruolo, la richiesta viene rifiutata. John passa due tag di sessione nella richiesta. Imposta questi due tag come transitivi. La richiesta eredita il tag `Department` come transitivo perché John lo ha impostato come transitivo quando ha assunto `JohnRole1`. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti. Per ulteriori informazioni sulle chiavi transitive nelle catene di ruoli, consulta Concatenamento dei ruoli con i tag di sessione.

```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIN5ATK5U7KEXAMPLE:JohnRole1",
        "arn": "arn:aws:sts::111111111111:assumed-role/JohnDoe/JohnRole1",
        "accountId": "111111111111",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-10-02T21:50:54Z"
            },
                "sessionIssuer": {
                "type": "Role",
               "principalId": "AROAIN5ATK5U7KEXAMPLE",
                "arn": "arn:aws:iam::111111111111:role/JohnRole1",
                "accountId": "111111111111",
                "userName": "JohnDoe"
            },
            "sourceIdentity": "JohnDoe"
        }
    },
    "eventTime": "2019-10-02T22:12:29Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "123.145.67.89",
    "userAgent": "aws-cli/1.16.248 Python/3.4.7
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 botocore/1.12.239",
    "requestParameters": {
        "incomingTransitiveTags": {
            "Department": "Engineering"
        },
        "tags": [
            {
                "value": "johndoe@example.com",
                "key": "Email"
            },
            {
                "value": "12345",
                "key": "CostCenter"
            }
        ],
        "roleArn": "arn:aws:iam::111111111111:role/JohnRole2",
```

```
        "roleSessionName": "Role2WithTags",
        "sourceIdentity": "JohnDoe",
        "transitiveTagKeys": [
            "Email",
            "CostCenter"
        ],
        "durationSeconds": 3600
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
            "expiration": "Oct 2, 2019, 11:12:29 PM",
            "sessionToken": "AgoJb3JpZ2luX2VjEB4aCXVzLXdlc3QtMSJHMEXAMPLETOKEN
+//rJb8Lo30mFc5MlhFCEbubZvEj0wHB/mDMwIgSEe9gk/Zjr09tZV7F1HDTMhmEXAMPLETOKEN/iEJ/
rkqngII9//////////
ARABGgw0MjgzMDc4NjM5NjYiDLZjZFKwP4qxQG5sFCryASO4UPz5qE97wPPH1eLMvs7CgSDBSWfonmRTCfokm2FN1+hWUdQ
+C+WKFZb701eiv9J5La2EXAMPLETOKEN/c7S5Iro1WUJ0q3Cxuo/8HUoSxVhQHM7zF7mWWLhXLEQ52ivL
+F6q5dpXu4aTFedpMfnJa8JtkWwG9x1Axj0Ypy2ok8v5unpQGWych1vwdvj6ez1Dm8Xg1+qIzXILiEXAMPLETOKEN/
vQGqu8H+nxp3kabcrtOvTFTvxX6vsc8OGwUfHhzAfYGEXAMPLETOKEN/
L6v1yMM3B1OwFOrQBno1HEjf1oNI8RnQiMNFdUOtwYj7HUZIOCZmjfN8PPHq77N7GJl9lzvIZKQA0Owcjg
+mc78zHCj8y0siY8C96paEXAMPLETOKEN/
E3cpksxWdgs91HRzJWScjN2+r2LTGjYhyPqcmFzzo2mCE7mBNEXAMPLETOKEN/oJy
+2o83YNW5tOiDmczgDzJZ4UKR84yGYOMfSnF4XcEJrDgAJ3OJFwmTcTQICAlSwLEXAMPLETOKEN"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAIFR7WHDTSOYQYHFUE:Role2WithTags",
            "arn": "arn:aws:sts::111111111111:assumed-role/test-role/Role2WithTags"
        },
    "sourceIdentity": "JohnDoe"
    },
    "requestID": "b96b0e4e-e561-11e9-8b3f-7b396EXAMPLE",
    "eventID": "1917948f-3042-46ec-98e2-62865EXAMPLE",
    "resources": [
        {
            "ARN": "arn:aws:iam::111111111111:role/JohnRole2",
            "accountId": "111111111111",
            "type": "AWS::IAM::Role"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```

Esempio AWS di evento AWS STS API di servizio nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata da un AWS servizio che chiama un'altra API di servizio utilizzando le autorizzazioni di un ruolo di servizio. Mostra la voce di CloudTrail registro per la richiesta effettuata nell'account 777788889999.


```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAQRSTUVWXYZEXAMPLE:devdsk",
    "arn": "arn:aws:sts::777788889999:assumed-role/AssumeNothing/devdsk",
    "accountId": "777788889999",
    "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-11-14T17:25:26Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAQRSTUVWXYZEXAMPLE",
        "arn": "arn:aws:iam::777788889999:role/AssumeNothing",
        "accountId": "777788889999",
        "userName": "AssumeNothing"
      }
    }
  },
  "eventTime": "2016-11-14T17:25:45Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "DeleteBucket",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "[aws-cli/1.11.10 Python/2.7.8
 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67]",
  "requestParameters": {
    "bucketName": "my-test-bucket-cross-account"
  },
  "responseElements": null,
  "requestID": "EXAMPLE463D56D4C",
  "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "777788889999"
```

```
}
```

Esempio di evento AWS STS API SAML nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l' AWS STS `AssumeRoleWithSAML`azione. La richiesta include gli attributi SAML `CostCenter` e `Project` che vengono passati tramite l'asserzione SAML come tag di sessione. Tali tag sono impostati come transitivi in modo da garantirne la persistenza negli scenari di concatenazione del ruolo. La richiesta include il parametro API opzionale`DurationSeconds`, rappresentato `durationSeconds` nel CloudTrail registro, ed è impostata su `1800` secondi. La richiesta include anche l'attributo SAML `sourceIdentity`, che viene inviato nell'asserzione SAML. Se un utente utilizza le credenziali della sessione del ruolo risultante per assumere un altro ruolo, questa identità di origine persiste.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "SAMLUser",
        "principalId": "SampleUkh1i4+ExamplexL/jEvs=:SamlExample",
        "userName": "SamlExample",
        "identityProvider": "bdGOnTesti4+ExamplexL/jEvs="
    },
    "eventTime": "2023-08-28T18:30:58Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoleWithSAML",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "aws-internal/3 aws-sdk-java/1.12.479
 Linux/5.10.186-157.751.amzn2int.x86_64 OpenJDK_64-Bit_Server_VM/17.0.7+11 java/17.0.7
 kotlin/1.3.72 vendor/Amazon.com_Inc. cfg/retry-mode/standard",
    "requestParameters": {
        "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
        "roleSessionName": "MyAssignedRoleSessionName",
        "sourceIdentity": "MySAMLUser",
        "principalTags": {
            "CostCenter": "987654",
            "Project": "Unicorn",
            "Department": "Engineering"
        },
        "transitiveTagKeys": [
            "CostCenter",
            "Project"
        ],
```

```
        "roleArn": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth",
        "principalArn": "arn:aws:iam::444455556666:saml-provider/Shibboleth",
        "durationSeconds": 1800
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "<encoded session token blob>",
            "expiration": "Aug 28, 2023, 7:00:58 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAD35QRSTUVWEXAMPLE:MyAssignedRoleSessionName",
            "arn": "arn:aws:sts::444455556666:assumed-role/SAMLTestRoleShibboleth/
MyAssignedRoleSessionName"
        },
        "packedPolicySize": 1,
        "subject": "SamlExample",
        "subjectType": "transient",
        "issuer": "https://server.example.com/idp/shibboleth",
        "audience": "https://signin.aws.amazon.com/saml",
        "nameQualifier": "bdGOnTesti4+ExamplexL/jEvs=",
        "sourceIdentity": "MySAMLUser"
    },
    "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
    "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth"
        },
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::SAMLProvider",
            "ARN": "arn:aws:iam::444455556666:saml-provider/test-saml-provider"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "444455556666",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
```

```
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "sts.us-east-2.amazonaws.com"
    }
}
```

Esempio di evento AWS STS API OIDC nel CloudTrail file di registro

L'esempio seguente mostra una voce di CloudTrail registro per una richiesta effettuata per l'azione AWS STS `AssumeRoleWithWebIdentity`. La richiesta include gli attributi `CostCenter` e `Project` che vengono passati tramite il token di un provider di identità come tag di sessione. Tali tag sono impostati come transitivi in modo da garantirne la persistenza negli scenari di concatenazione del ruolo. La richiesta include l'attributo `sourceIdentity` dal token del provider di identità. Se un utente utilizza le credenziali della sessione del ruolo risultante per assumere un altro ruolo, questa identità di origine persiste.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "WebIdentityUser",
    "principalId": "accounts.google.com:<id-of-
application>.apps.googleusercontent.com:<id-of-user>",
    "userName": "<id of user>",
    "identityProvider": "accounts.google.com"
  },
  "eventTime": "2016-03-23T01:39:51Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRoleWithWebIdentity",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5",
  "requestParameters": {
    "sourceIdentity": "MyWebIdentityUser",
    "durationSeconds": 3600,
    "roleArn": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
    "roleSessionName": "MyAssignedRoleSessionName"
        "principalTags": {
            "CostCenter": "24680",
            "Project": "Pegasus"
        },
        "transitiveTagKeys": [
            "CostCenter",
            "Project"
```

```
        ],
  },
  "responseElements": {
    "provider": "accounts.google.com",
    "subjectFromWebIdentityToken": "<id of user>",
    "sourceIdentity": "MyWebIdentityUser",
    "audience": "<id of application>.apps.googleusercontent.com",
    "credentials": {
      "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
      "expiration": "Mar 23, 2016, 2:39:51 AM",
      "sessionToken": "<encoded session token blob>"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AROACQRSTUVWRAOEXAMPLE:MyAssignedRoleSessionName",
      "arn": "arn:aws:sts::444455556666:assumed-role/FederatedWebIdentityRole/
MyAssignedRoleSessionName"
    }
  },
  "resources": [
    {
      "ARN": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
      "accountId": "444455556666",
      "type": "AWS::IAM::Role"
    }
  ],
  "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
  "eventID": "bEXAMPLE-0b30-4246-b28c-e3da3EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "444455556666"
}
```

Esempio di eventi di accesso nel log CloudTrail

CloudTrail i file di registro contengono eventi formattati utilizzando JSON. Un evento di accesso rappresenta una singola richiesta di accesso e include informazioni sul principale di accesso, la regione e la data e l'ora dell'operazione.

Esempio di evento di accesso riuscito nel file di registro CloudTrail

L'esempio seguente mostra una voce di CloudTrail registro per un evento di accesso riuscito.

```
{
  "eventVersion": "1.05",
```

```
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn":"arn:aws:iam::111122223333:user/JohnDoe",
    "accountId": "111122223333",
    "userName": "JohnDoe"
  },
  "eventTime": "2014-07-16T15:49:27Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.110",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Success"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "LoginTo": "https://console.aws.amazon.com/s3/  ",
    "MFAUsed": "No"
  },
  "eventID": "3fcfb182-98f8-4744-bd45-10a395ab61cb"
}
```


Per ulteriori dettagli sulle informazioni contenute nei file di CloudTrail registro, vedere CloudTrail Event Reference nella Guida per l'AWS CloudTrail utente.

Esempio di errore di accesso nel CloudTrail file di registro

L'esempio seguente mostra una voce di CloudTrail registro per un evento di accesso non riuscito.


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn":"arn:aws:iam::111122223333:user/JaneDoe",
    "accountId": "111122223333",
    "userName": "JaneDoe"
  },
  "eventTime": "2014-07-08T17:35:27Z",
```

```
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.100",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "errorMessage": "Failed authentication",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "LoginTo": "https://console.aws.amazon.com/sns",
    "MFAUsed": "No"
  },
  "eventID": "11ea990b-4678-4bcd-8fbe-62509088b7cf"
}
```

Da queste informazioni puoi determinare che il tentativo di accesso è stato effettuato da un utente IAM denominato `JaneDoe`, come riportato nell'elemento `userIdentity`. Puoi anche consultare che il tentativo di accesso non è riuscito, come indicato nell'elemento `responseElements`. Puoi verificare che `JaneDoe` ha provato ad accedere alla console Amazon SNS alle 17:35 UTC in data 8 luglio 2014.

Esempio di evento di accesso non riuscito a causa di un nome utente non corretto

L'esempio seguente mostra una voce di CloudTrail registro relativa a un evento di accesso non riuscito causato dall'immissione di un nome utente errato da parte dell'utente. AWS maschera il `userName` testo con lo scopo `HIDDEN_DUE_TO_SECURITY_REASONS` di impedire la divulgazione di informazioni potenzialmente sensibili.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "accountId": "123456789012",
    "accessKeyId": "",
    "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "eventTime": "2015-03-31T22:20:42Z",
  "eventSource": "signin.amazonaws.com",
```

```
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101
 Firefox/24.0",
  "errorMessage": "No username found in supplied account",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
    "MobileVersion": "No",
    "MFAUsed": "No"
  },
  "eventID": "a7654656-0417-45c6-9386-ea8231385051",
  "eventType": "AwsConsoleSignin",
  "recipientAccountId": "123456789012"
}
```

IAM role trust policy, comportamento

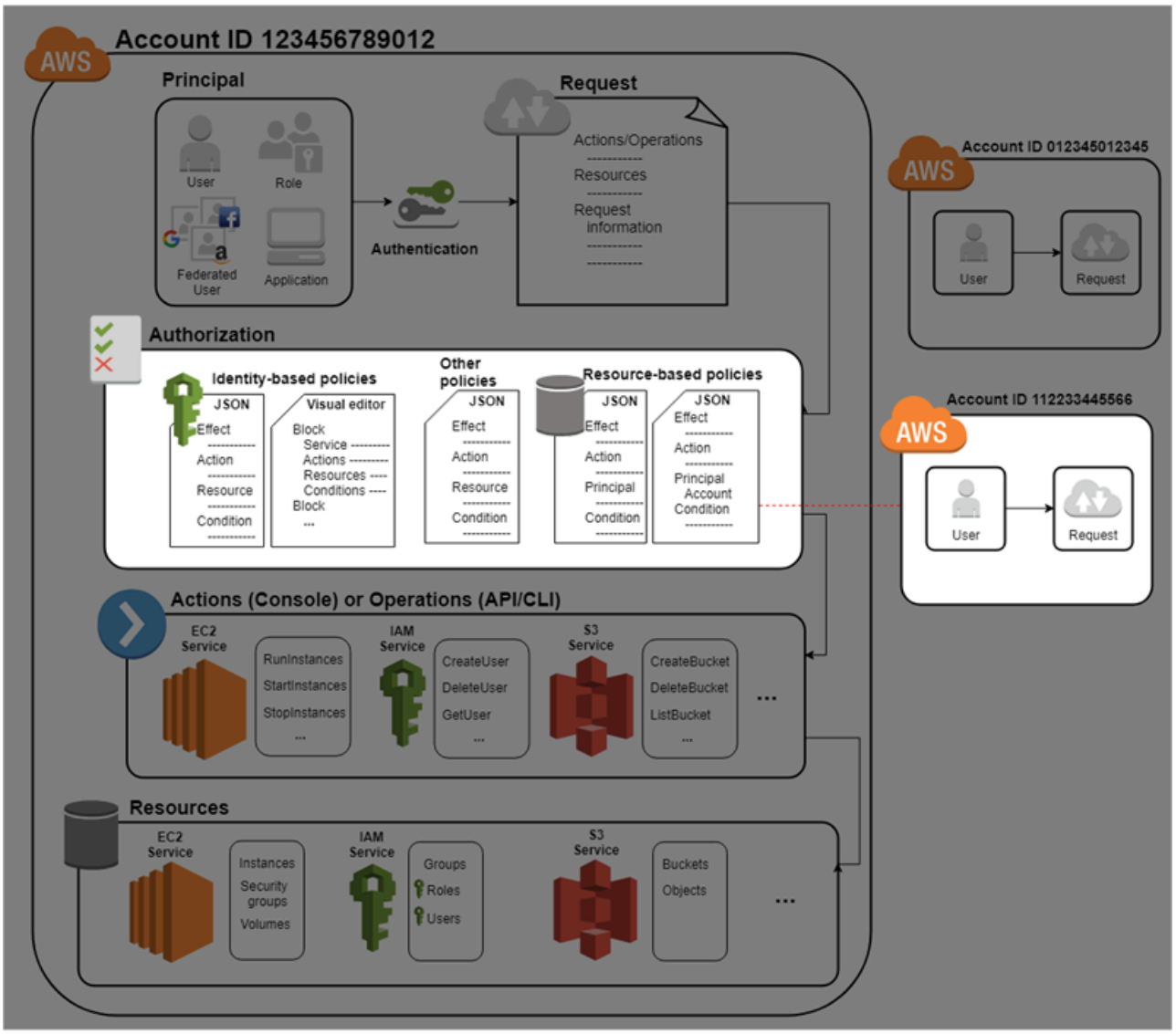
Il 21 settembre 2022, AWS ha apportato modifiche al comportamento della policy di trust dei ruoli IAM per richiedere una politica esplicita di autorizzazioni in un ruolo quando un ruolo si assume da solo. I ruoli IAM nella precedente lista dei comportamenti consentiti hanno un additionalEventData campo presente explicitTrustGrant per `AssumeRole` gli eventi. Il valore di `explicitTrustGrant` è falso quando un ruolo nella precedente lista consentita si presume di utilizzare il comportamento legacy. Quando un ruolo nella precedente lista consentita assume se stesso ma il comportamento della politica di fiducia dei ruoli è stato aggiornato per consentire esplicitamente al ruolo di assumere se stesso, il valore di `explicitTrustGrant` è true.

Solo un numero molto limitato di ruoli IAM è presente nell'elenco dei ruoli consentiti per il comportamento legacy e questo campo è presente nei CloudTrail log di questi ruoli solo quando assumono se stessi. Nella maggior parte dei casi, non è necessario che un ruolo IAM assuma se stesso. AWS consiglia di aggiornare i processi, il codice o le configurazioni per rimuovere questo comportamento o di aggiornare le policy di fiducia dei ruoli per consentire esplicitamente questo comportamento. Per ulteriori informazioni, consulta Annuncio di un aggiornamento del comportamento delle policy di fiducia dei ruoli di IAM.

Gestione degli accessi AWS alle risorse

AWS Identity and Access Management (IAM) è un servizio web che consente di controllare in modo sicuro l'accesso alle AWS risorse. Quando un principale effettua una richiesta di ingresso AWS, il codice di AWS applicazione verifica se il principale è autenticato (registrato) e autorizzato (dispone delle autorizzazioni). Puoi gestire l'accesso AWS creando policy e collegandole a identità o risorse IAM. AWS Le policy sono documenti JSON AWS che, se allegate a un'identità o a una risorsa, ne definiscono le autorizzazioni. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Policy e autorizzazioni in IAM.

Per ulteriori informazioni sul resto del processo di autenticazione e autorizzazione, consultare Funzionamento di IAM.



Durante l'autorizzazione, il codice di AWS applicazione utilizza i valori del contesto della richiesta per verificare le politiche corrispondenti e determinare se consentire o rifiutare la richiesta.

AWS controlla ogni politica che si applica al contesto della richiesta. Se una singola politica nega la richiesta, AWS nega l'intera richiesta e interrompe la valutazione delle politiche. Questa azione si chiama rifiuto esplicito. Poiché le richieste vengono rifiutate per impostazione predefinita, IAM autorizza la richiesta solo se ogni parte di essa è autorizzata dalle policy applicabili. La logica di valutazione per una richiesta all'interno di un singolo account segue queste regole:

- Per impostazione predefinita, tutte le richieste vengono negate implicitamente (in alternativa, per impostazione predefinita, l' Utente root dell'account AWS ha accesso completo).
- Un'autorizzazione esplicita in una policy basata su identità o basata su risorse sostituisce questa impostazione predefinita.
- Se è presente un limite delle autorizzazioni, una SCP di Organizations oppure una policy di sessione, potrebbe sovrascrivere l'autorizzazione con un rifiuto implicito.
- Un rifiuto esplicito in una policy sostituisce qualsiasi permesso.

Dopo che la richiesta è stata autenticata e autorizzata, AWS approva la richiesta. Se hai bisogno di effettuare una richiesta in un altro account, una policy nell'altro account deve consentire l'accesso alla risorsa. Inoltre, l'entità IAM utilizzata per effettuare la richiesta deve avere una policy basata su identità che consente la richiesta.

Accesso alle risorse di gestione

Per ulteriori informazioni sulle autorizzazioni e sulla creazione di policy, consultare le seguenti risorse:

Le seguenti voci del AWS Security Blog descrivono i modi più comuni per scrivere politiche per l'accesso a bucket e oggetti Amazon S3.

- Scrittura di policy IAM: come concedere l'accesso a un bucket Amazon S3
- Scrittura di policy IAM: concessione dell'accesso a cartelle specifiche dell'utente in un bucket Amazon S3
- Policy IAM, policy di bucket e ACL Oh My! (Controllo dell'accesso alle risorse S3)
- Un primer sulle autorizzazioni a livello di risorsa RDS
- Demifisticazione delle autorizzazioni a livello di risorsa EC2

Policy e autorizzazioni in IAM

Puoi gestire l'accesso AWS creando policy e collegandole a identità o risorse IAM (utenti, gruppi di utenti o ruoli). AWS Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un responsabile IAM (utente o ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata AWS come documenti JSON. AWS supporta sei tipi di policy: policy basate sull'identità, policy basate sulle risorse, permessi limitati, Organizations SCP, ACL e policy di sessione.

Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, se una policy consente l'GetUserazione, un utente con quella policy può ottenere informazioni sull'utente dall'API o dall'API. AWS Management Console AWS CLI AWS Nella creazione di un utente IAM, è possibile scegliere di consentire l'accesso programmatico o alla console. Se è consentito l'accesso alla console, l'utente IAM può accedere alla console con le proprie credenziali di accesso. Se è consentito l'accesso programmatico, l'utente può utilizzare le chiavi di accesso per utilizzare la CLI o l'API.

Tipi di policy

I tipi di policy elencati di seguito in ordine da quello utilizzato più frequentemente a quello meno frequentemente sono disponibili per l'uso in AWS. Per ulteriori informazioni, consulta le sezioni seguenti per ogni tipo di policy.

- Policy basate su identità: collega le policy gestite e inline alle identità IAM (utenti, gruppi a cui appartengono gli utenti o ruoli). Le policy basate su identità concedono le autorizzazioni a un'identità.
- Policy basate sulle risorse: collegano le policy in linea alle risorse. Gli esempi più comuni di policy basate su risorse sono le policy dei bucket Amazon S3 e le policy di attendibilità dei ruoli IAM. Le policy basate su risorse concedono le autorizzazioni a un'identità principale specificata nella policy. Le entità principali possono essere nello stesso account della risorsa o in altri account.
- Limiti delle autorizzazioni: utilizza una policy gestita come limite delle autorizzazioni per un'entità IAM (utente o ruolo). Questa policy definisce il numero massimo di autorizzazioni che la policy basata su identità può concedere a un'entità, ma non concede autorizzazioni. I limiti delle autorizzazioni non definiscono il numero massimo di autorizzazioni che una policy basata su risorse può concedere a un'entità.

- Organizations SCP: utilizza una policy di controllo dei AWS Organizations servizi (SCP) per definire le autorizzazioni massime per i membri dell'account di un'organizzazione o di un'unità organizzativa (OU). Le SCP limitano le autorizzazioni che le policy basate su identità o le policy basate su risorse concedono alle entità (utenti o ruoli) all'interno dell'account, ma non concedono autorizzazioni.
- Liste di controllo accessi (ACL): utilizza le ACL per controllare quali entità principali in altri account possono accedere alla risorsa a cui è collegata l'ACL. Le ACL sono simili alle policy basate su risorse, anche se sono l'unico tipo di policy che non utilizza la struttura del documento di policy JSON. Le ACL sono policy di autorizzazione tra più account che concedono le autorizzazioni all'identità principale specificata. Le ACL non possono concedere autorizzazioni a entità nello stesso account.
- Criteri di sessione: applica criteri di sessione avanzati quando utilizzi l' AWS API AWS CLI o per assumere un ruolo o un utente federato. Le policy di sessione limitano le autorizzazioni che le policy basate su identità dell'utente o del ruolo concedono alla sessione. Le policy di sessione limitano le autorizzazioni per una sessione creata, ma non possono concedere autorizzazioni. Per ulteriori informazioni, consulta la sezione relativa alle policy di sessione.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che controllano quali operazioni un'identità (utenti, gruppi di utenti e ruoli) può eseguire, su quali risorse e in quali condizioni. Le policy basate su identità possono essere ulteriormente suddivise:

- Politiche gestite: politiche autonome basate sull'identità che puoi allegare a più utenti, gruppi e ruoli nel tuo. Account AWS Sono disponibili due tipi di policy gestite.
  - AWS politiche gestite: politiche gestite create e gestite da. AWS
  - Policy gestite dal cliente: le policy gestite che sono create e gestite nel tuo Account AWS. Le politiche gestite dal cliente forniscono un controllo più preciso sulle politiche rispetto alle politiche AWS gestite.
- Policy in linea: le policy che vengono aggiunte direttamente a un singolo utente, gruppo o ruolo. Le politiche in linea mantengono una stretta one-to-one relazione tra una politica e un'identità. Vengono eliminate quando elimini l'identità.

Per informazioni su come scegliere tra una policy gestita o una policy in linea, consulta Scelta fra policy gestite e policy inline.

Policy basate su risorse

Le policy basate sulle risorse sono documenti di policy JSON che colleghi a una risorsa, come ad esempio un bucket Amazon S3. Queste policy concedono all'entità principale specificata l'autorizzazione per eseguire operazioni specifiche sulla risorsa e definiscono le condizioni in cui ciò si applica. Le policy basate su risorse sono policy inline. Non esistono policy basate su risorse gestite.

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono separati Account AWS, è inoltre necessario utilizzare una politica basata sull'identità per concedere l'accesso principale alla risorsa. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per istruzioni dettagliate sulla concessione dell'accesso tra servizi, consulta Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.

Il servizio IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Un ruolo IAM è sia un'identità che una risorsa che supporta le policy basate sulle risorse. Per questo motivo, è necessario collegare sia una policy di attendibilità sia una policy basata su identità a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Per capire in che modo i ruoli IAM si differenziano da altre policy basate su risorse, consulta Accesso alle risorse multi-account in IAM.

Per scoprire quali altri servizi supportano le policy basate su risorse, consulta AWS servizi che funzionano con IAM. Per ulteriori informazioni sulle policy basate su risorse, consulta la pagina Policy basate sulle identità e policy basate su risorse. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

Limiti delle autorizzazioni IAM

Un limite delle autorizzazioni è una funzione avanzata in cui si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Quando si imposta un limite delle autorizzazioni per un'entità, l'entità può eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo come entità principale non sono limitate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.

Policy di controllo dei servizi (Service Control Policies, SCP)

AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata dei dati di proprietà dell'azienda Account AWS . Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. Le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o un'unità organizzativa (UO). L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

Per ulteriori informazioni su Organizations e le SCP, consulta Utilizzo delle SCP nella Guida per l'utente di AWS Organizations .

Liste di controllo degli accessi (ACL)

Le liste di controllo accessi (ACL) sono policy di servizio che consentono di controllare quali principali in un altro account possono accedere a una risorsa. Le ACL non possono essere utilizzate per controllare l'accesso per un'entità principale all'interno dello stesso account. Le ACL sono simili alle policy basate su risorse, anche se sono l'unico tipo di policy che non utilizza il formato del documento di policy JSON. Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica della lista di controllo accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Policy di sessione

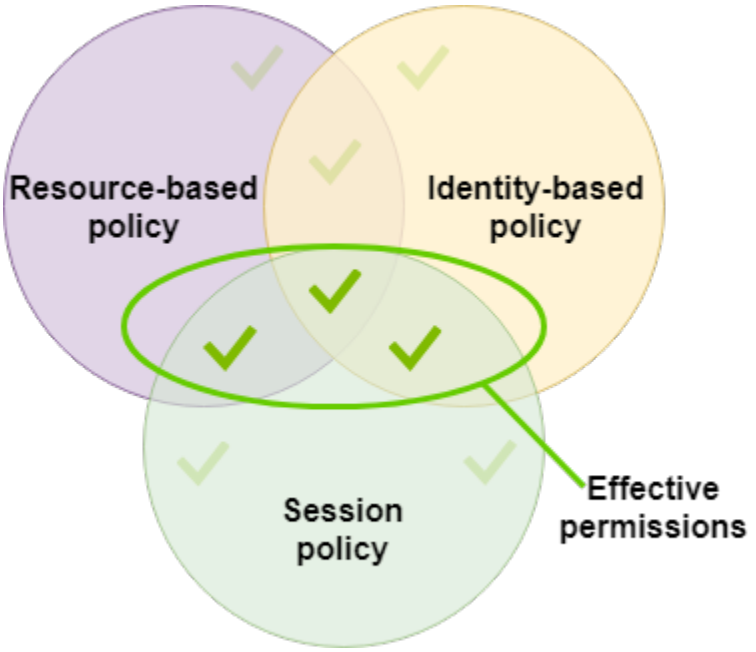
Le policy di sessione sono policy avanzate che si passano come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni per una sessione sono l'intersezione delle policy basate su identità per l'entità IAM (utente o ruolo) utilizzate per creare la sessione e delle policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

Puoi creare una sessione del ruolo e passare policy di sessione a livello di programmazione utilizzando le operazioni API `AssumeRole`, `AssumeRoleWithSAML` o `AssumeRoleWithWebIdentity`. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Per ulteriori informazioni sulla creazione di una sessione del ruolo, consulta Richiesta di credenziali di sicurezza temporanee.

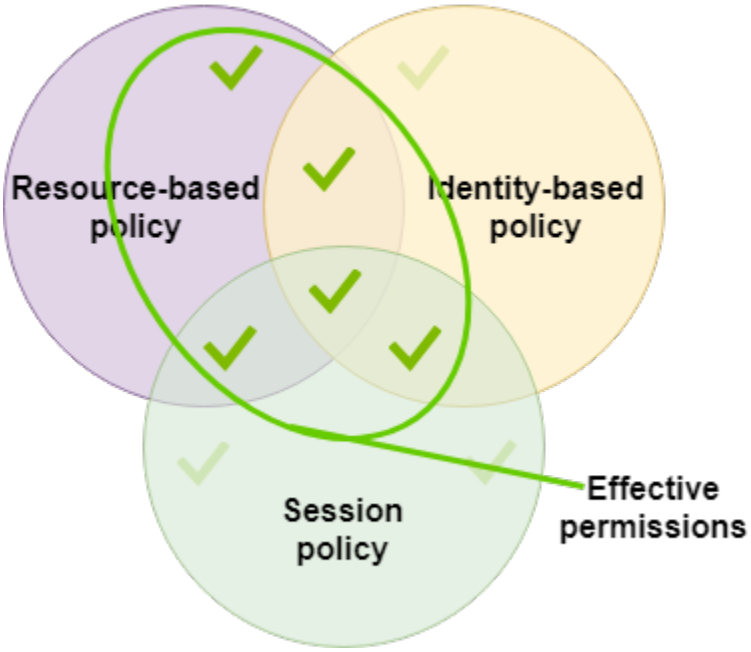
Quando crei una sessione per l'utente federato, utilizzi le chiavi di accesso dell'utente IAM per chiamare in modo programmatico l'operazione API `GetFederationToken`. È inoltre necessario

passare policy di sessione. Le autorizzazioni della sessione risultanti sono l'intersezione tra la policy basata su identità e la policy di sessione. Per ulteriori informazioni sulla creazione di una sessione per l'utente federato, consulta GetFederationToken: federazione tramite un gestore identità personalizzato.

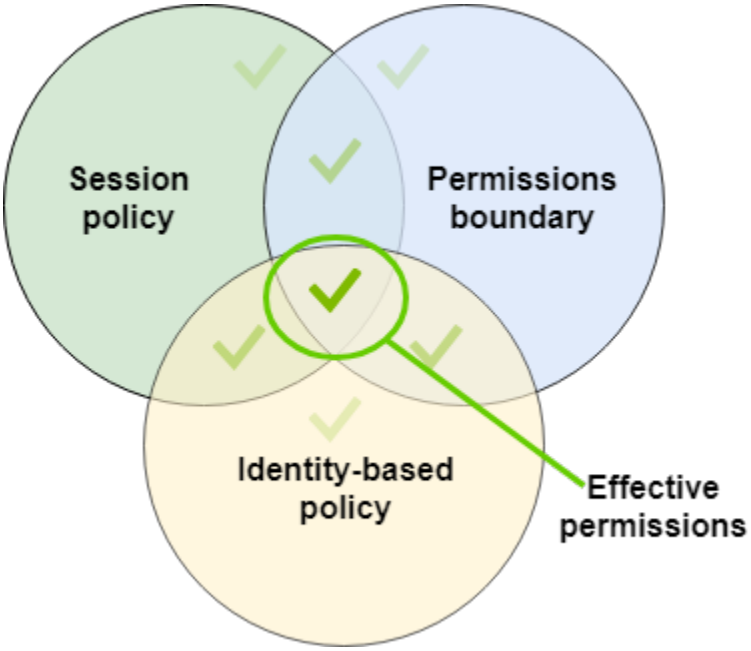
Una policy basata sulle risorse è in grado di specificare l'ARN dell'utente o del ruolo come un principale. In questo caso, le autorizzazioni della policy basata sulle risorse vengono aggiunte alla policy basata su identità dell'utente o del ruolo prima che la sessione venga creata. La policy di sessione limita le autorizzazioni totali concesse dalla policy basata sulle risorse e dalla policy basata su identità. Le autorizzazioni della sessione risultante sono l'intersezione delle policy di sessione e della policy basata sulle risorse più l'intersezione delle policy di sessione e delle policy basate su identità.



Una policy basata sulle risorse è in grado di specificare l'ARN della sessione come un principale. In questo caso, le autorizzazioni della policy basata sulle risorse vengono aggiunte dopo che la sessione viene creata. Le autorizzazioni della policy basate sulle risorse non sono limitate dalla policy di sessione. La sessione risultante dispone di tutte le autorizzazioni della policy basata sulle risorse più l'intersezione della policy basata su identità e della policy di sessione.



Un limite delle autorizzazioni è in grado di impostare il numero massimo di autorizzazioni per un utente o un ruolo che viene utilizzato per creare una sessione. In tal caso, le autorizzazioni della sessione risultante sono l'intersezione della policy di sessione, il limite delle autorizzazioni e la policy basata su identità. Tuttavia, un limite delle autorizzazioni non limita le autorizzazioni concesse da una policy basata sulle risorse che specifica l'ARN della sessione risultante.



Policy e utente root

Utente root dell'account AWS È influenzato da alcuni tipi di policy ma non da altri. Non è possibile collegare policy basate su identità all'utente root e non è possibile impostare il limite delle autorizzazioni per questo utente. Tuttavia, è possibile specificare l'utente root come principale in una policy basata su risorse o un'ACL. Un utente root è ancora membro di un account. Se quell'account è membro di un'organizzazione in AWS Organizations, l'utente root è interessato dagli SCP dell'account.

Panoramica delle policy JSON

La maggior parte delle politiche viene archiviata AWS come documenti JSON. Le policy basate su identità e quelle utilizzate per impostare i limiti delle autorizzazioni sono documenti di policy JSON che vengono collegati a un utente o un ruolo. Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli SCP sono documenti di policy JSON con sintassi limitata allegati a un'unità AWS Organizations organizzativa (OU). Anche le ACL vengono collegate a una risorsa, ma è necessario utilizzare una sintassi differente. Le policy di sessione sono le policy JSON fornite quando si assume un ruolo o una sessione per l'utente federato.

Non è necessario conoscere la sintassi JSON. È possibile utilizzare l'editor visivo in AWS Management Console per creare e modificare le politiche gestite dai clienti senza mai utilizzare JSON. Tuttavia, se utilizzi policy inline per gruppi o policy complesse, devi comunque creare e modificare tali policy nell'editor JSON tramite la console. Per ulteriori informazioni sull'uso dell'editor grafico, consultare Creazione di policy IAM e Modifica delle policy IAM.

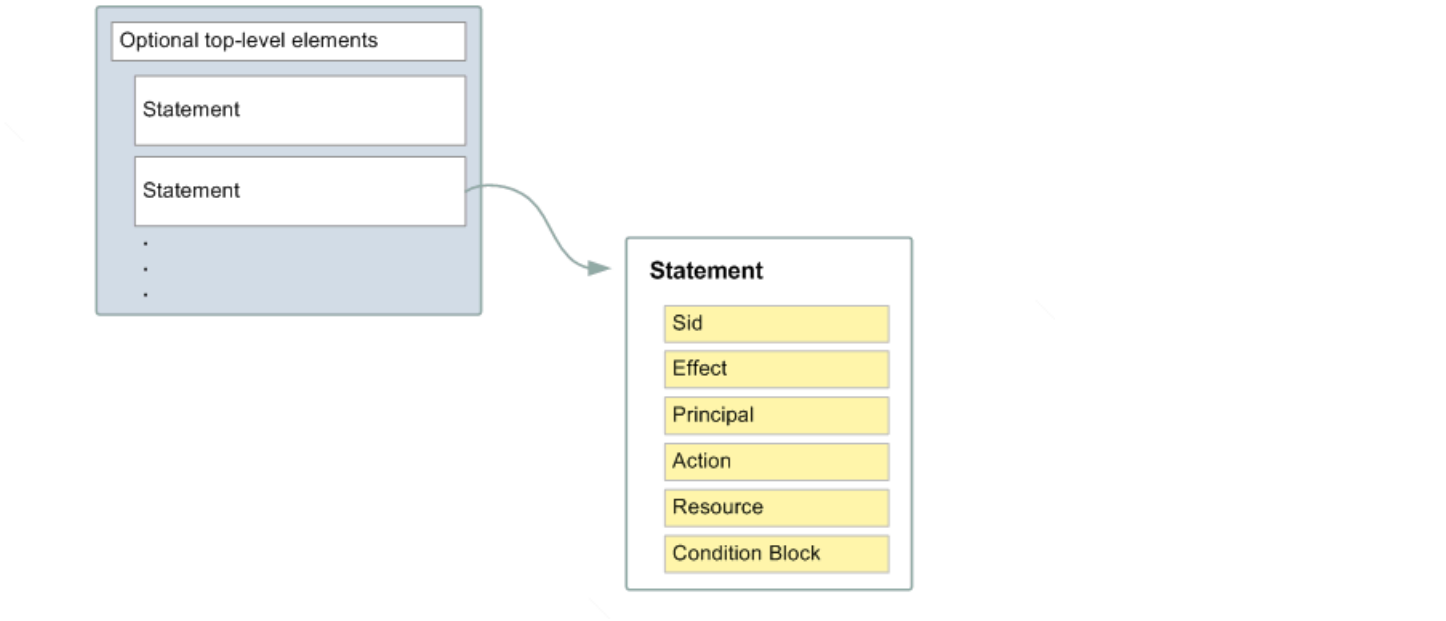
Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Struttura dei documenti di policy JSON

Come illustrato nella figura di seguito, un documento di policy JSON include questi elementi:

- Informazioni opzionali sulla policy nella parte superiore del documento
- Una o più istruzioni singole

Ogni istruzione include informazioni su una singola autorizzazione. Se una politica include più istruzioni, AWS applica una logica a `OR` tutte le istruzioni durante la valutazione. Se a una richiesta si applicano più politiche, AWS applica una logica a `OR` tutte quelle politiche durante la valutazione.



Le informazioni di un'istruzione sono contenute all'interno di una serie di elementi.

- Version: specifica la versione del linguaggio di policy che desideri utilizzare. Consigliamo di utilizzare la versione `2012-10-17` più recente. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Version
- Statement: utilizza questo elemento principale della policy come container per i seguenti elementi. Puoi includere più istruzioni in una policy.
- Sid (facoltativo): includi un ID istruzione opzionale per distinguere le varie istruzioni.
- Effect: utilizza `Allow` o `Deny` per indicare se la policy consente l'accesso o lo rifiuta.
- Principal (obbligatorio solo in alcune circostanze): se crei una policy basata sulle risorse, devi indicare l'account, l'utente, il ruolo o l'utente federato a cui desideri consentire o rifiutare l'accesso. Nella creazione di una policy di autorizzazioni IAM da collegare a un utente o un ruolo, non è possibile includere questo elemento. L'entità principale è implicita come l'utente o il ruolo.
- Action: includi un elenco delle operazioni consentite o rifiutate dalla policy.
- Resource (obbligatorio solo in alcune circostanze): se crei una policy di autorizzazioni IAM, devi specificare un elenco di risorse a cui si applicano le operazioni. Se crei una policy basata sulle risorse, questo elemento è facoltativo. Se non includi questo elemento, la risorsa a cui si applica l'operazione è la risorsa a cui è collegata la policy.

- Condition (facoltativo): specifica le circostanze in base alle quali la policy concede l'autorizzazione.

Per informazioni su questi e altri elementi di policy più avanzati, consulta Documentazione di riferimento degli elementi delle policy JSON IAM.

Istruzioni e policy multiple

Per definire più di un'autorizzazione per un'entità (utente o ruolo), puoi utilizzare più istruzioni in una singola policy. Puoi anche collegare più policy. Se tenti di definire più autorizzazioni in un'unica istruzione, la policy potrebbe non concedere l'accesso come previsto. Ti consigliamo di suddividere le policy in base al tipo di risorsa.

A causa delle dimensioni limitate delle policy, può essere necessario utilizzare più policy per le autorizzazioni più complesse. È inoltre consigliabile creare raggruppamenti funzionali di autorizzazioni in una policy gestita dal cliente separata. Ad esempio, crea una policy per la gestione degli utenti IAM, una per la gestione automatica e un'altra per la gestione dei bucket S3. Indipendentemente dalla combinazione di più dichiarazioni e più politiche, AWS valuta le politiche allo stesso modo.

Ad esempio, la policy seguente include tre istruzioni, ciascuna delle quali definisce un set di autorizzazioni separato all'interno di un unico account. Le istruzioni definiscono quanto segue:

- La prima istruzione, con un `Sid` (ID istruzione) di `FirstStatement`, consente all'utente con la policy collegata di modificare la propria password. In questa istruzione l'elemento `Resource` è `*` (che significa "tutte le risorse"). Tuttavia, in pratica, l'operazione API `ChangePassword` (o il comando CLI `change-password` equivalente) influisce solo sulla password per l'utente che effettua la richiesta.
- La seconda istruzione consente all'utente di elencare tutti i bucket Amazon S3 del proprio Account AWS. In questa istruzione l'elemento `Resource` è `"*"` (che significa "tutte le risorse"). Tuttavia, poiché le policy non concedono l'accesso alle risorse di altri account, l'utente può elencare solo i bucket del proprio Account AWS.
- La terza istruzione consente all'utente di elencare e recuperare qualsiasi oggetto all'interno di un bucket denominato `confidential-data`, ma solo quando l'utente viene autenticato con la multi-factor authentication (MFA). L'elemento `Condition` della policy applica l'autenticazione MFA.

  Quando un'istruzione della policy contiene un elemento `Condition`, l'istruzione risulta valida solo se per l'elemento `Condition` viene restituito un valore True. In questo caso, `Condition` restituisce True se l'utente è stato autenticato mediante MFA. Se l'utente non dispone

dell'autenticazione MFA, `Condition` restituisce False. In tal caso, la terza istruzione della policy non è applicabile e l'utente non può accedere al bucket `confidential-data`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FirstStatement",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword"],
      "Resource": "*"
    },
    {
      "Sid": "SecondStatement",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "ThirdStatement",
      "Effect": "Allow",
      "Action": [
        "s3:List*",
        "s3:Get*"
      ],
      "Resource": [
        "arn:aws:s3:::confidential-data",
        "arn:aws:s3:::confidential-data/*"
      ],
      "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
    }
  ]
}
```

Esempi di sintassi di policy JSON

La policy basata sulle identità riportata di seguito consente all'entità principale implicita di elencare un singolo bucket Amazon S3 denominato `example_bucket`:

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

La policy basata su risorse riportata di seguito può essere collegata a un bucket Amazon S3. La policy consente ai membri di uno specifico utente di Account AWS eseguire qualsiasi azione di Amazon S3 nel bucket denominato. `mybucket` Consente qualsiasi operazione che possa essere eseguita su un bucket o sugli oggetti in esso contenuti. Poiché la policy concede la fiducia solo agli account, i singoli utenti dell'account dovranno comunque ricevere le autorizzazioni per le operazioni Amazon S3 specificate.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::account-id:root"]},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }]
}
```

Per alcuni esempi di policy per scenari comuni, consulta Esempi di policy basate su identità IAM.

Assegnare il privilegio minimo

Quando crei le policy IAM, segui i consigli di sicurezza standard sulla concessione di privilegi minimi o sulla concessione delle sole autorizzazioni richieste per eseguire un'attività. Determina i compiti di utenti e ruoli, quindi crea policy che consentono loro di eseguire solo tali attività.

Inizia con un set di autorizzazioni minimo e concedi autorizzazioni aggiuntive quando necessario. Questo è più sicuro che iniziare con autorizzazioni che siano troppo permissive e cercare di limitarle in un secondo momento.

In alternativa al minimo privilegio, puoi usare le autorizzazioni di policy gestite da AWS o policy con carattere jolly * per iniziare a utilizzare le policy. Considera il rischio per la sicurezza di concedere ai

principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro. Monitora tali principali per sapere quali autorizzazioni stanno utilizzando. Quindi scrivi le policy con il privilegio minimo.

IAM fornisce diverse opzioni che consentono di perfezionare le autorizzazioni concesse.

- Informazioni sui raggruppamenti a livello di accesso: puoi utilizzare i raggruppamenti a livello di accesso per comprendere il livello di accesso concesso da una policy. Le operazioni delle policy sono classificate come `List`, `Read`, `Write`, `Permissions management` o `Tagging`. Ad esempio, è possibile selezionare operazioni dai livelli di accesso `List` e `Read` per concedere accesso in sola lettura agli utenti. Per ulteriori informazioni su come utilizzare i riepiloghi delle policy per comprendere le autorizzazioni a livello di accesso, consultare Comprensione dei livelli di accesso nei riepiloghi delle politiche.
- Convalida le policy: puoi eseguire la convalida delle policy utilizzando IAM Access Analyzer quando crei e modifichi le policy JSON. Consigliamo di rivedere e convalidare tutte le policy esistenti. Per convalidare le policy, IAM Access Analyzer fornisce oltre 100 controlli delle policy. Genera avvisi di sicurezza quando una istruzione nella tua policy consente l'accesso che consideriamo eccessivamente permissivo. È possibile utilizzare i suggerimenti utili forniti tramite gli avvisi di sicurezza mentre si lavora per concedere il minimo privilegio. Per ulteriori informazioni sui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.
- Genera una policy basata sull'attività di accesso: per ottimizzare le autorizzazioni concesse, puoi generare una policy IAM basata sull'attività di accesso per un'entità IAM (utente o ruolo). IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nel periodo di tempo specificato. È possibile utilizzare il modello per creare una policy gestita con autorizzazioni granulari e quindi collegarla al ruolo IAM. In questo modo, concedi solo le autorizzazioni necessarie all'utente o al ruolo per interagire con le AWS risorse per il tuo caso d'uso specifico. Per ulteriori informazioni, consulta Generazione di policy basate sull'attività di accesso.
- Utilizza informazioni sull'ultimo accesso: un'altra funzionalità che può aiutarti con il minimo privilegio èInformazioni sull'ultimo accesso. Visualizza queste informazioni nella scheda Access Advisor nella pagina dei dettagli della console IAM per un utente, un gruppo, un ruolo o una policy IAM. Le informazioni sull'ultimo accesso includono anche informazioni sulle azioni a cui si è effettuato l'ultimo accesso per alcuni servizi, ad esempio Amazon EC2, IAM, Lambda e Amazon S3. Se accedi utilizzando le credenziali dell'account di AWS Organizations gestione, puoi visualizzare le informazioni sull'ultimo accesso al servizio nella AWS Organizationssezione della console IAM. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare un report con le informazioni dell'ultimo accesso per entità o policy in IAM o Organizations. Puoi utilizzare queste

informazioni per identificare le autorizzazioni non necessarie, in modo da perfezionare le policy IAM o Organizations per aderire meglio al principio del privilegio minimo. Per ulteriori informazioni, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

- Rivedi gli eventi dell'account in AWS CloudTrail: per ridurre ulteriormente le autorizzazioni, puoi visualizzare gli eventi del tuo account nella AWS CloudTrail Cronologia degli eventi. CloudTrail i registri degli eventi includono informazioni dettagliate sugli eventi che puoi utilizzare per ridurre le autorizzazioni della politica. I log includono solo le operazioni e le risorse richieste dalle entità IAM. Per ulteriori informazioni, vedere Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Per ulteriori informazioni, consulta i seguenti argomenti di policy per singoli servizi, che forniscono esempi di come scrivere policy per le risorse specifiche del servizio.

- Autenticazione e controllo degli accessi per Amazon DynamoDB nella Guida per gli sviluppatori di Amazon DynamoDB
- Utilizzo delle policy di bucket e delle policy utente nella Guida per l'utente di Amazon Simple Storage Service.
- Panoramica sulla lista di controllo accessi (ACL) nella Guida per l'utente di Amazon Simple Storage Service

Policy gestite e policy inline

Quando imposti le autorizzazioni per un'identità in IAM, dovrai scegliere tra una policy gestita da AWS , una policy gestita dal cliente o una policy inline. Gli argomenti seguenti forniscono ulteriori informazioni su ciascuno dei tipi di policy basate sull'identità e su quando utilizzarli.

Argomenti

- AWS politiche gestite
- Policy gestite dal cliente
- Policy inline
- Scelta fra policy gestite e policy inline
- Nozioni di base sulle policy gestite
- Conversione di una policy inline in una policy gestita

- Policy gestite obsolete AWS

AWS politiche gestite

Una policy gestita da AWS è una policy autonoma che viene creata e amministrata da AWS. Policy autonoma significa che la policy ha un proprio Amazon Resource Name (ARN) che include il nome della policy. Ad esempio, `arn:aws:iam::aws:policy/IAMReadOnlyAccess` è una politica AWS gestita. Per ulteriori informazioni sugli ARN, consultare la pagina ARN IAM. Per un elenco delle politiche AWS gestite per Servizi AWS, vedere Politiche AWS gestite.

AWS le politiche gestite semplificano l'assegnazione delle autorizzazioni appropriate a utenti, gruppi e ruoli. È più veloce della scrittura delle policy in autonomia e include le autorizzazioni per molti casi d'uso comuni.

Non è possibile modificare le autorizzazioni definite nelle AWS politiche gestite. AWS aggiorna occasionalmente le autorizzazioni definite in una politica AWS gestita. In tal caso AWS , l'aggiornamento influisce su tutte le entità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando viene lanciato un nuovo AWS servizio o quando nuove chiamate API diventano disponibili per i servizi esistenti. Ad esempio, la policy AWS gestita denominata ReadOnlyAccessfornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando AWS avvia un nuovo servizio, AWS aggiorna la ReadOnlyAccesspolitica per aggiungere autorizzazioni di sola lettura per il nuovo servizio. Le autorizzazioni aggiornate vengono applicate a tutte le entità principali a cui la policy è collegata.

Le politiche di accesso completo AWS gestite definiscono le autorizzazioni per gli amministratori del servizio concedendo l'accesso completo a un servizio.

- AmazonDynamoDB FullAccess
- IAM FullAccess

Le policy AWS gestite dagli utenti esperti forniscono l'accesso completo a AWS servizi e risorse, ma non consentono la gestione di utenti e gruppi.

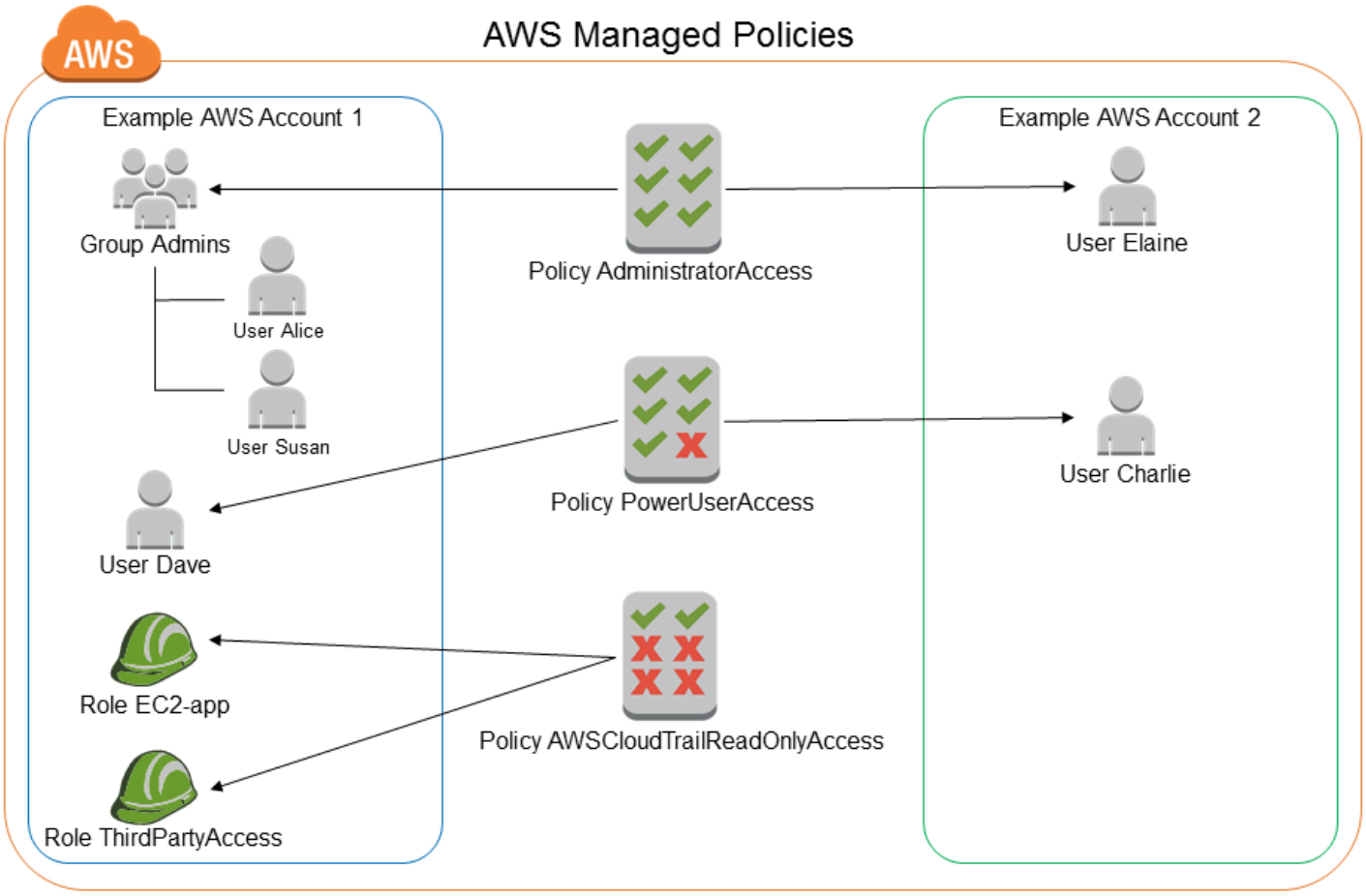
- AWSCodeCommitPowerUser
- AWSKeyManagementServicePowerUser

Le policy AWS gestite ad accesso parziale forniscono livelli specifici di accesso ai AWS servizi senza consentire la gestione delle autorizzazioni (autorizzazioni a livello di accesso).

- AmazonMobileAnalyticsWriteOnlyAccess
- Amazon EC2 ReadOnlyAccess

Una categoria particolarmente utile di politiche AWS gestite sono quelle progettate per le funzioni lavorative. Queste policy si allineano strettamente alle funzioni lavorative comunemente utilizzate nel settore IT e facilitano la concessione delle autorizzazioni per queste funzioni lavorative. Uno dei principali vantaggi dell'utilizzo delle politiche relative alle funzioni lavorative è che vengono mantenute e aggiornate AWS man mano che vengono introdotti nuovi servizi e operazioni API. Ad esempio, la funzione AdministratorAccessjob fornisce l'accesso completo e la delega delle autorizzazioni a ogni servizio e risorsa in AWS uso. Si consiglia di utilizzare questa policy solo per l'amministratore dell'account. Per gli utenti esperti che richiedono l'accesso completo a tutti i servizi tranne l'accesso limitato a IAM and Organizations, utilizza la funzione PowerUserAccessjob. Per un elenco e descrizioni delle policy delle mansioni lavorative, consulta AWS politiche gestite per le funzioni lavorative.

Il diagramma seguente illustra le politiche AWS gestite. Il diagramma mostra tre politiche AWS gestite: AdministratorAccess, e PowerUserAccess. AWS CloudTrailReadOnlyAccess Si noti che una singola politica AWS gestita può essere associata a entità principali in diverse Account AWS entità principali e a diverse entità principali in un'unica Account AWS entità.



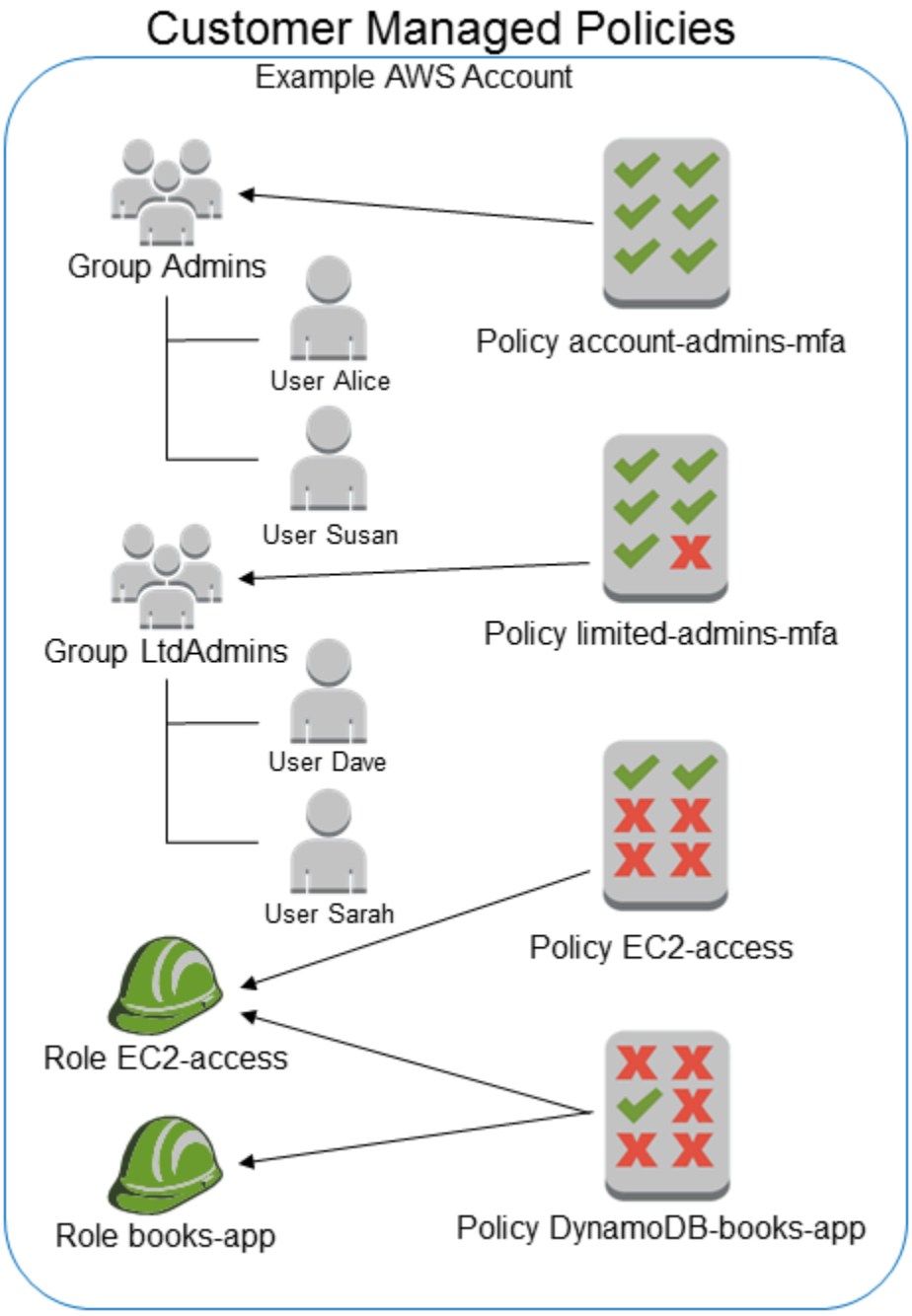
Policy gestite dal cliente

È possibile creare politiche autonome personalizzate Account AWS da allegare alle entità principali (utenti, gruppi e ruoli). Puoi creare queste policy gestite dal cliente per i tuoi casi d'uso specifici e modificarle e aggiornarle tutte le volte che desideri. AWS Analogamente alle politiche gestite, quando si allega una politica a un'entità principale, si assegnano all'entità le autorizzazioni definite nella politica. Quando le autorizzazioni della policy vengono aggiornate, le modifiche vengono applicate a tutte le entità principali a cui è collegata la policy.

Un ottimo modo per creare una policy gestita dal cliente è iniziare copiando una policy gestita da AWS esistente. In questo modo è possibile assicurarsi che la policy sia corretta come base ed è sufficiente personalizzarla per il proprio ambiente.

Il diagramma seguente illustra le policy gestite dal cliente. Ogni policy è un'entità in IAM con un proprio Amazon Resource Name (ARN) che include il nome della policy. Si noti che la stessa policy può essere collegata a più entità principali, ad esempio, la stessa policy DynamoDB-books-app è collegata a due diversi ruoli IAM.

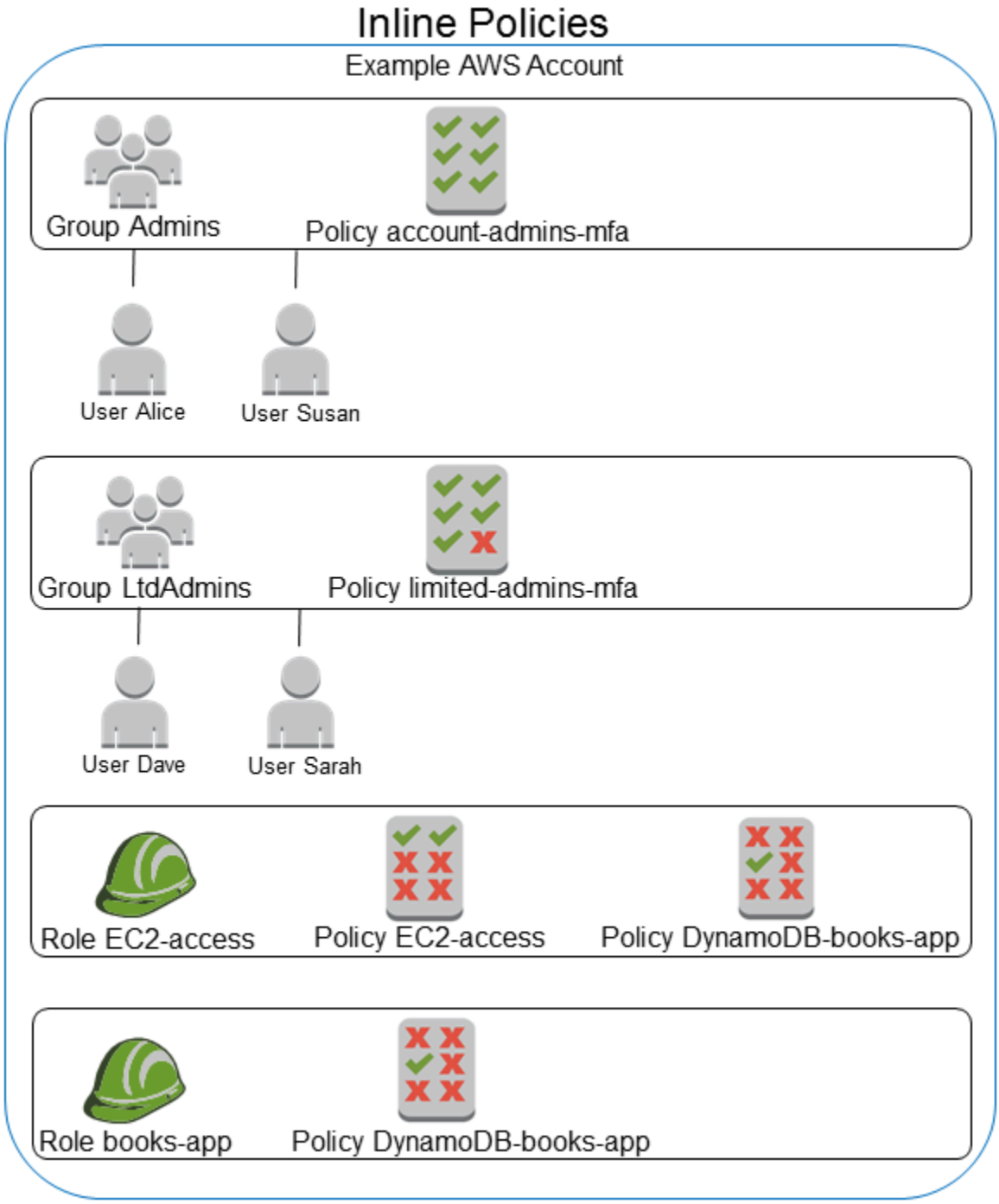
Per ulteriori informazioni, consulta Creazione di policy IAM



Policy inline

Una policy inline è una policy creata per una singola identità IAM (utente, gruppo o ruolo). Le politiche in linea mantengono una stretta one-to-one relazione tra una politica e un'identità. Vengono eliminate quando elimini l'identità. È possibile creare una policy e incorporarla in un'identità, sia quando si crea l'identità sia in un secondo momento. Se una policy può essere applicata a più di un'entità, è meglio utilizzare una policy gestita.

Il diagramma seguente illustra le policy inline. Ogni policy è parte integrante dell'utente, gruppo o ruolo. Si noti che i due ruoli includono la stessa policy (la policy DynamoDB-books-app), ma non condividono una singola policy. Ogni ruolo dispone di una propria copia della policy.



Scelta fra policy gestite e policy inline

Al momento di scegliere tra policy gestite e policy inline, prendi in considerazione i casi d'uso. Nella maggior parte dei casi, si consiglia di usare le policy gestite anziché le policy inline.

> **Note**
>
> È possibile utilizzare insieme le policy gestite e policy inline per definire autorizzazioni comuni e univoche per un'entità principale.

Le policy gestite offrono le seguenti caratteristiche:

Riutilizzo

Una singola policy gestita può essere collegata a più entità principali (utenti, gruppi e ruoli). È possibile creare una libreria di politiche che definiscono le autorizzazioni utili per l'utente Account AWS e quindi allegarle alle entità principali in base alle esigenze.

Gestione centralizzata delle modifiche

Quando si modifica una policy gestita, la modifica viene applicata a tutte le entità principali a cui la policy è collegata. Ad esempio, se desideri aggiungere l'autorizzazione per una nuova AWS API, puoi aggiornare una politica gestita dal cliente o associare una politica AWS gestita per aggiungere l'autorizzazione. Se utilizzi una policy AWS gestita, AWS aggiorna la policy. Quando una politica gestita viene aggiornata, le modifiche vengono applicate a tutte le principali entità a cui è associata la politica gestita. Al contrario, per modificare una politica in linea, è necessario modificare singolarmente ogni identità che contiene la politica in linea. Ad esempio, se un gruppo e un ruolo contengono la stessa policy inline, è necessario modificare individualmente entrambe le entità principali per modificare tale policy.

Controllo delle versioni e rollback

Quando si modifica una policy gestita dal cliente, la policy modificata non sovrascriverà la policy esistente. IAM crea invece una nuova versione della policy gestita. IAM memorizza fino a cinque versioni di una policy gestita dal cliente. È possibile utilizzare le versioni della policy per ripristinare una policy a una versione precedente, se necessario.

> **Note**
>
> Una versione di policy è diversa da un elemento `Version` della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Per ulteriori informazioni sulle versioni di policy, consultare the section called “Controllo delle versioni delle policy IAM”. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version.

Delega della gestione delle autorizzazioni

Puoi consentire agli utenti del tuo account Account AWS di allegare e scollegare le policy mantenendo il controllo sulle autorizzazioni definite in tali politiche. A tale scopo, è possibile designare alcuni utenti come amministratori completi, ossia amministratori che possono creare, aggiornare ed eliminare le policy. È quindi possibile designare altri utenti come amministratori limitati. Tali amministratori limitati possono collegare delle policy ad altre entità principali, ma solo nel caso delle policy per le quali sono stati autorizzati.

Per ulteriori informazioni sulla delega della gestione delle autorizzazioni, consultare Controllo dell'accesso alle policy.

Limiti di caratteri per le policy più grandi

Il limite massimo di caratteri per le policy gestite è maggiore del limite di caratteri per le policy in linea. Se raggiungi il limite di dimensione dei caratteri della policy in linea, puoi creare altri gruppi IAM e collegare la policy gestita al gruppo.

Per ulteriori informazioni su quote e limiti, consulta IAM e AWS STS quote.

Aggiornamenti automatici per AWS le politiche gestite

AWS mantiene le politiche AWS gestite e le aggiorna quando necessario, ad esempio per aggiungere autorizzazioni per nuovi AWS servizi, senza che l'utente debba apportare modifiche. Gli aggiornamenti vengono applicati automaticamente alle principali entità a cui è stata allegata la politica AWS gestita.

Utilizzo delle policy inline

Le politiche in linea sono utili se si desidera mantenere una one-to-one relazione stretta tra una politica e l'identità a cui viene applicata. Ad esempio, se si desidera essere certi che le autorizzazioni di una policy non vengano inavvertitamente assegnate a un'identità diversa da quella per la quale

sono state concepite. Quando si utilizza una policy inline, le autorizzazioni della policy non possono essere collegate inavvertitamente a un'identità errata. Inoltre, quando si utilizza il AWS Management Console per eliminare tale identità, vengono eliminate anche le politiche incorporate nell'identità perché fanno parte dell'entità principale.

Nozioni di base sulle policy gestite

Consigliamo di utilizzare le policy che concedono il privilegio minimo o che concedono solo le autorizzazioni richieste per eseguire un processo. Il modo più sicuro per concedere il privilegio minimo consiste nello scrivere una policy gestita dal cliente solo con le autorizzazioni necessarie al team. È necessario creare un processo per consentire al team di richiedere ulteriori autorizzazioni quando necessario. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza.

Per iniziare ad aggiungere autorizzazioni alle tue identità IAM (utenti, gruppi di utenti e ruoli), puoi usare. AWS politiche gestite AWS le politiche gestite non concedono i permessi con il privilegio minimo. Considera il rischio per la sicurezza di concedere ai principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro.

Puoi allegare policy AWS gestite, incluse le funzioni lavorative, a qualsiasi identità IAM. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Per passare alle autorizzazioni con privilegi minimi, puoi eseguire AWS Identity and Access Management Access Analyzer per monitorare i principali con policy gestite. AWS Dopo aver appreso quali autorizzazioni stanno utilizzando, puoi scrivere o generare una policy gestita dal cliente che contenga soltanto le autorizzazioni richieste per il team. È meno sicuro, ma offre maggiore flessibilità man mano che impari a utilizzare il tuo team. AWS Per ulteriori informazioni, consulta Generazione di policy per Sistema di analisi degli accessi AWS IAM.

AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni. Per ulteriori informazioni sulle politiche AWS gestite progettate per funzioni lavorative specifiche, vedereAWS politiche gestite per le funzioni lavorative.

Per un elenco delle politiche AWS gestite, consulta la AWS Managed Policy Reference Guide.

Conversione di una policy inline in una policy gestita

Se disponi di policy inline nell'account, puoi convertirle in policy gestite. A tale scopo, copia la policy in una nuova policy gestita, collega la nuova policy all'identità che ha la policy inline, quindi elimina la policy inline.

Per convertire una policy inline in una policy gestita

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
3. Nell'elenco, scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
4. Scegli la scheda Autorizzazioni.
5. Per i gruppi di utenti, seleziona il nome della policy in linea che desideri rimuovere. Per gli utenti e i ruoli, scegliere Mostra altri ***n***, se necessario, quindi espandi la policy inline da rimuovere.
6. Scegli Copia per copiare il documento della policy in formato JSON.
7. Nel riquadro di navigazione, seleziona Policy.
8. Seleziona Crea policy, quindi scegli l'opzione JSON.
9. Sostituisci il testo esistente con il testo della policy JSON, quindi scegli Verifica policy.
10. Immetti un nome e una descrizione facoltativa per la policy, quindi scegli Crea policy.
11. Nel pannello di navigazione, scegli Gruppi di utenti, Utenti o Ruoli e scegli di nuovo il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
12. Seleziona la scheda Autorizzazioni e scegli Aggiungi autorizzazioni.
13. Per i gruppi, seleziona la casella di controllo accanto al nome della nuova policy, quindi scegli Aggiungi autorizzazioni, quindi Collega policy. Per gli utenti o i ruoli, scegliere Add permissions (Aggiungi autorizzazioni). Nella pagina successiva, scegli Collega direttamente policy esistenti, seleziona la casella di controllo accanto al nome della nuova policy, scegli Successivo, quindi seleziona Aggiungi autorizzazioni.

    Sarai riportato alla pagina Riepilogo per l'utente, il gruppo di utenti o il ruolo.
14. Seleziona la casella di controllo accanto alla policy inline che desideri rimuovere, quindi scegli Rimuovi.

Policy gestite obsolete AWS

Per semplificare l'assegnazione delle autorizzazioni, AWS fornisce policy gestite, ossia policy predefinite pronte per essere associate agli utenti, ai gruppi e ai ruoli IAM.

A volte è AWS necessario aggiungere una nuova autorizzazione a una politica esistente, ad esempio quando viene introdotto un nuovo servizio. L'aggiunta di una nuova autorizzazione a una policy esistente non disturba o rimuove qualsiasi caratteristica o possibilità.

Tuttavia, AWS potrebbe scegliere di creare una nuova politica quando le modifiche necessarie potrebbero avere un impatto sui clienti se applicate a una politica esistente. Ad esempio, la rimozione delle autorizzazioni da una policy esistente potrebbe violare le autorizzazioni di qualsiasi entità o applicazione IAM dalla quale dipendeva, disturbando potenzialmente un'operazione critica.

Pertanto, quando è necessaria una tale modifica, AWS crea una politica completamente nuova con le modifiche richieste e la mette a disposizione dei clienti. La policy vecchia viene contrassegnata come obsoleta. Una policy gestita obsoleta appare con un'icona di avviso vicino all'elenco Policy nella console IAM.

Una policy obsoleta presenta le seguenti caratteristiche:

- Continua a funzionare per tutti gli utenti, gruppi e ruoli attualmente collegati. Nessuna interruzione.
- Non può essere collegata a nuovi utenti, gruppi e ruoli. Se viene scollegata da un'entità corrente, non è possibile collegarla nuovamente.
- Dopo averla scollegata da tutte le entità correnti, non è più visibile e non può essere utilizzata in alcun modo.

Se qualsiasi utente, gruppo o ruolo richiede la policy, bisogna invece collegare la nuova policy. Quando si riceve un avviso che una policy è obsoleta, è consigliabile collegare immediatamente tutti gli utenti, gruppi e ruoli per la policy di sostituzione e scollegarli dalla policy obsoleta. Continuare a utilizzare la policy obsoleta può creare rischi mitigati solo dal passaggio alla policy di sostituzione.

Stabilisci barriere di autorizzazione utilizzando i perimetri dei dati

Le barriere perimetrali dei dati sono pensate per fungere da confini permanenti per aiutare a proteggere i dati su un'ampia gamma di account e risorse. AWS I perimetri dei dati seguono le migliori pratiche di sicurezza IAM per stabilire barriere di autorizzazioni su più account. Queste barriere di autorizzazione a livello di organizzazione non sostituiscono i controlli di accesso dettagliati esistenti. Funzionano invece come controlli di accesso generalizzati che aiutano a migliorare la strategia di sicurezza assicurando che utenti, ruoli e risorse aderiscano a una serie di standard di sicurezza definiti.

Un perimetro di dati è un insieme di barriere di autorizzazione nell' AWS ambiente che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili delle reti previste.

- Identità affidabili: i responsabili (ruoli o utenti IAM) dei tuoi AWS account e servizi che agiscono per tuo conto. AWS

- Risorse affidabili: risorse di proprietà AWS dei tuoi account o di AWS servizi che agiscono per tuo conto.
- Reti previste: i data center locali e i cloud privati virtuali (VPC) o le reti di AWS servizi che agiscono per conto dell'utente.

> **Note**
>
> In alcuni casi, potrebbe essere necessario estendere il perimetro dei dati per includere anche l'accesso da parte dei partner commerciali di fiducia. È necessario prendere in considerazione tutti i modelli di accesso ai dati previsti quando si crea una definizione di identità affidabili, risorse attendibili e reti previste specifiche per l'azienda e l'utilizzo delle stesse. Servizi AWS

I controlli perimetrali dei dati devono essere trattati come qualsiasi altro controllo di sicurezza nell'ambito del programma di sicurezza delle informazioni e di gestione del rischio. Ciò significa che è necessario eseguire un'analisi delle minacce per identificare i potenziali rischi all'interno del proprio ambiente cloud e quindi, in base ai propri criteri di accettazione del rischio, selezionare e implementare controlli perimetrali dei dati appropriati. Per definire meglio l'approccio iterativo basato sul rischio all'implementazione del perimetro dei dati, è necessario comprendere quali rischi e vettori di minaccia vengono affrontati dai controlli perimetrali dei dati, nonché le priorità di sicurezza.

Controlli perimetrali dei dati

I controlli a grana grossa sul perimetro dei dati aiutano a raggiungere sei obiettivi di sicurezza distinti su tre perimetri di dati attraverso l'implementazione di diverse combinazioni di chiavi e condizioni. Tipi di policy

Perimetro	Obiettivo di controllo	Utilizzo	Applicato su	Chiavi contestuali delle condizioni globali
Identità	Solo le identità affidabili possono	Policy basata su risorse	Risorse	aws: ID Principal Org leggi: Principal OrgPaths

Perimetro	Obiettivo di controllo	Utilizzo	Applicato su	Chiavi contestuali delle condizioni globali
	accedere alle mie risorse			seghe: PrincipalAccount
	Nella mia rete sono consentite solo identità affidabili	Policy degli endpoint VPC	Rete	seghe: PrincipalIsAwsService
Risorse	Le tue identità possono accedere solo a risorse affidabili	SCP	Identità	aws: IDResourceOrg leggi: ResourceOrgPaths seghe: ResourceAccount
	Dalla rete è possibile accedere solo a risorse affidabili	Policy degli endpoint VPC	Rete	
Rete	Le tue identità possono accedere alle risorse solo dalle reti previste	SCP	Identità	leggi: SourceIp seghe: SourceVpc seghe: SourceVpce AWS: via AWSService leggi: PrincipalIsAwsService
	È possibile accedere alle risorse solo dalle reti previste	Policy basata su risorse	Risorse	

Puoi pensare ai perimetri dei dati come alla creazione di un confine preciso attorno ai tuoi dati per prevenire schemi di accesso non intenzionali. Sebbene i perimetri dei dati possano impedire

ampi accessi involontari, è comunque necessario prendere decisioni granulari sul controllo degli accessi. La definizione di un perimetro di dati non riduce la necessità di ottimizzare continuamente le autorizzazioni utilizzando strumenti come IAM Access Analyzer come parte del percorso verso il privilegio minimo.

Perimetro di identità

Un perimetro di identità è un insieme di controlli preventivi di accesso generalizzati che aiutano a garantire che solo le identità attendibili possano accedere alle risorse e che solo le identità affidabili siano consentite dalla rete. Le identità affidabili includono i responsabili (ruoli o utenti) degli account e dei servizi che agiscono per conto dell'utente. AWS AWS Tutte le altre identità sono considerate non attendibili e sono impedite dal perimetro dell'identità, a meno che non venga concessa un'eccezione esplicita.

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali delle identità. Utilizza queste chiavi nelle policy basate sulle risorse per limitare l'accesso alle risorse o nelle policy degli endpoint VPC per limitare l'accesso alle tue reti.

- aws: PrincipalOrg ID— È possibile utilizzare questa chiave di condizione per garantire che i responsabili IAM che effettuano la richiesta appartengano all'organizzazione specificata in. AWS Organizations
- aws: PrincipalOrg Percorsi— È possibile utilizzare questa chiave di condizione per garantire che l'utente IAM, il ruolo IAM, l'utente federato o A che Utente root dell'account AWS effettua la richiesta appartengano all'unità organizzativa (OU) specificata in. AWS Organizations
- leggi: PrincipalAccount— È possibile utilizzare questa chiave di condizione per garantire che le risorse siano accessibili solo all'account principale specificato nella politica.
- leggi: PrincipalIs AWSServicee aws: SourceOrg ID (alternativamente aws: SourceOrg Percorsi eleggi: SourceAccount): è possibile utilizzare queste chiavi di condizione per garantire che, quando Servizio AWS i responsabili accedono alle risorse, lo facciano solo per conto di una risorsa dell'organizzazione, dell'unità organizzativa o di un account in specificata. AWS Organizations

Per ulteriori informazioni, consulta Stabilire un perimetro di dati su AWS: Consenti solo alle identità affidabili di accedere ai dati aziendali.

Perimetro delle risorse

Un perimetro di risorse è un insieme di controlli di accesso preventivi di accesso generalizzati che aiutano a garantire che le identità possano accedere solo a risorse affidabili e che solo risorse

attendibili siano accessibili dalla rete. Le risorse affidabili includono risorse di proprietà degli AWS account o dei servizi che agiscono per conto dell'utente. AWS

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali delle risorse. Utilizza queste chiavi nelle policy di controllo dei servizi (SCP) per limitare le risorse a cui possono accedere le tue identità o nelle politiche degli endpoint VPC per limitare le risorse a cui è possibile accedere dalle tue reti.

- Leggi: ResourceOrg ID— È possibile utilizzare questa chiave di condizione per garantire che la risorsa a cui si accede appartenga all'organizzazione specificata in. AWS Organizations
- aws: ResourceOrg Percorsi— È possibile utilizzare questa chiave di condizione per garantire che la risorsa a cui si accede appartenga all'unità organizzativa (OU) specificata in AWS Organizations.
- Leggi: ResourceAccount— È possibile utilizzare questa chiave di condizione per garantire che la risorsa a cui si accede appartenga all'account specificato in AWS Organizations.

In alcuni casi, potrebbe essere necessario consentire l'accesso a risorse di AWS proprietà, risorse che non appartengono all'organizzazione e a cui accedono i responsabili o i AWS servizi che agiscono per conto dell'utente. Per ulteriori informazioni su questi scenari, consulta Stabilire un perimetro di dati su AWS: Consenti solo risorse attendibili della mia organizzazione.

Perimetro di rete

Un perimetro di rete è un insieme di controlli preventivi di accesso dettagliati che aiutano a garantire che le identità possano accedere alle risorse solo dalle reti previste e che le risorse siano accessibili solo dalle reti previste. Le reti previste includono i data center locali, i cloud privati virtuali (VPC) e le reti di servizi che agiscono per conto dell'utente. AWS

Le seguenti chiavi di condizione globali aiutano a far rispettare i controlli perimetrali della rete. Utilizzate queste chiavi nelle politiche di controllo dei servizi (SCP) per limitare le reti da cui le identità possono comunicare o nelle politiche basate sulle risorse per limitare l'accesso alle risorse alle reti previste.

- leggi: SourceIp— È possibile utilizzare questa chiave di condizione per garantire che l'indirizzo IP del richiedente rientri in un intervallo IP specificato.
- come: SourceVpc— È possibile utilizzare questa chiave di condizione per garantire che l'endpoint VPC attraverso cui viaggia la richiesta appartenga al VPC specificato.
- leggi: SourceVpce— È possibile utilizzare questa chiave di condizione per garantire che la richiesta viaggi attraverso l'endpoint VPC specificato.

- AWS: via AWSService— È possibile utilizzare questa chiave condizionale per assicurarsi che sia Servizi AWS possibile effettuare richieste per conto del proprio utente principale Inoltro delle sessioni di accesso (FAS).
- leggi: PrincipalIs AWSService— È possibile utilizzare questa chiave di condizione per assicurarsi che sia Servizi AWS possibile accedere alle risorse utilizzandoAWS presidi del servizio.

Esistono altri scenari in cui è necessario consentire l'accesso a Servizi AWS tale accesso alle risorse dall'esterno della rete. Per ulteriori informazioni, consulta Stabilire un perimetro di dati su AWS: Consenti l'accesso ai dati aziendali solo dalle reti previste.

Risorse per saperne di più sui perimetri dei dati

Le seguenti risorse possono aiutarti a saperne di più sui perimetri dei dati in tutto il mondo. AWS

- Perimetri dei dati attivi AWS: scopri i perimetri dei dati e i relativi vantaggi e casi d'uso.
- Whitepaper: Building a Data Perimeter on — AWS Questo paper delinea le migliori pratiche e i servizi disponibili per creare un perimetro attorno a identità, risorse e reti. AWS
- Webinar: Creazione di un perimetro di dati in: scopri dove e come implementare i controlli perimetrali dei dati in AWS base a diversi scenari di rischio.
- Serie di post sul blog: Stabilire un perimetro di dati su AWS — Questi post del blog forniscono linee guida prescrittive sulla definizione del perimetro dei dati su larga scala, comprese considerazioni chiave sulla sicurezza e l'implementazione.
- Esempi di policy relative al perimetro dei dati: questo GitHub repository contiene esempi di policy che coprono alcuni modelli comuni per aiutarti a implementare un perimetro di dati. AWS
- Supporto per il perimetro dei dati: questo strumento consente di progettare e anticipare l'impatto dei controlli perimetrali dei dati analizzando l'attività di accesso nei log. AWS CloudTrail

Limiti delle autorizzazioni per le entità IAM

AWS supporta i limiti delle autorizzazioni per le entità IAM (utenti o ruoli). Un limite delle autorizzazioni è una funzione avanzata per l'utilizzo di una policy gestita per impostare il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Il limite delle autorizzazioni di un'entità consente di eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni.

Per ulteriori informazioni sui tipi di policy, consulta Tipi di policy.

> **Important**
>
> Non utilizzare istruzioni di policy basate sulle risorse che includono un elemento di policy `NotPrincipal` con effetto `Deny` per gli utenti o i ruoli IAM ai quali è collegata una policy con limite delle autorizzazioni. L'elemento `NotPrincipal` con effetto `Deny` rifiuterà sempre qualsiasi principale IAM al quale è collegata una policy con limite delle autorizzazioni, indipendentemente dai valori specificati nell'elemento `NotPrincipal`. Ciò fa sì che alcuni utenti o ruoli IAM che altrimenti avrebbero accesso alla risorsa perdano l'accesso. Ti consigliamo di modificare le istruzioni di policy basate sulle risorse di modo che, per limitare l'accesso, utilizzino l'operatore di condizione `ArnNotEquals` con la chiave di contesto `aws:PrincipalArn` anziché l'elemento `NotPrincipal`. Per ulteriori informazioni sull'elemento `NotPrincipal`, consulta la pagina AWS Elementi della policy JSON: NotPrincipal.

Puoi utilizzare una policy AWS gestita o una policy gestita dal cliente per impostare il limite per un'entità IAM (utente o ruolo). La policy limita il numero massimo di autorizzazioni per l'utente o il ruolo.

Ad esempio, supponiamo che l'utente IAM denominato `ShirleyRodriguez` debba essere autorizzato a gestire solo Amazon S3 CloudWatch, Amazon e Amazon EC2. Per applicare la regola, puoi utilizzare la policy seguente per impostare il limite delle autorizzazioni per l'utente `ShirleyRodriguez`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Quando utilizzi una policy per impostare il limite delle autorizzazioni per un utente, questa limita le autorizzazioni dell'utente, ma non le fornisce di per sé. In questo esempio, la policy imposta le autorizzazioni massime di tutte `ShirleyRodriguez` le operazioni in Amazon S3 CloudWatch e Amazon EC2. Shirley non può eseguire operazioni negli altri servizi, incluso IAM, anche se dispone di una policy di autorizzazione che lo consente. Ad esempio, prova ad aggiungere la policy seguente all'utente `ShirleyRodriguez`:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

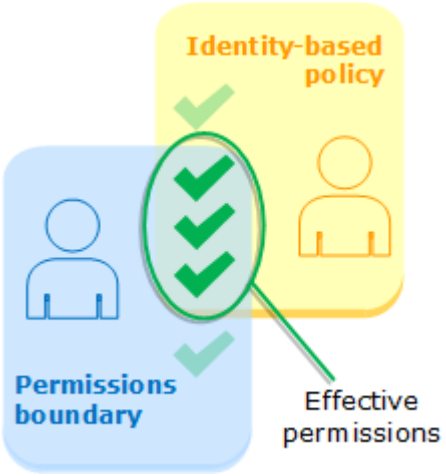
Questa policy consente la creazione di un utente in IAM. Se colleghi questa policy di autorizzazione all'utente `ShirleyRodriguez` e Shirley tenta di creare un utente, l'operazione ha esito negativo. Non riesce perché il limite delle autorizzazioni non consente l'operazione `iam:CreateUser`. Date queste due policy, Shirley non ha il permesso di eseguire alcuna operazione in AWS. È necessario aggiungere una policy di autorizzazioni diversa per consentire operazioni in altri servizi, ad esempio Amazon S3. In alternativa, è possibile aggiornare il limite delle autorizzazioni per consentirle di creare un utente in IAM.

Valutazione delle autorizzazioni valide con i limiti

Il limite delle autorizzazioni per un'entità IAM (utente o ruolo) imposta il numero massimo di autorizzazioni che è possibile concedere all'entità. Questo può influire sulle autorizzazioni valide per l'utente o il ruolo. Le autorizzazioni valide per un'entità sono quelle concesse da tutte le policy che interessano l'utente o il ruolo. In un account, le autorizzazioni per un'entità possono essere influenzate da policy basate su identità, policy basate su risorse, limiti delle autorizzazioni, SCP di Organizations o policy di sessione. Per ulteriori informazioni sui diversi tipi di policy, consulta Policy e autorizzazioni in IAM.

Se uno di questi tipi di policy rifiuta esplicitamente l'accesso per un'operazione, la richiesta viene rifiutata. Le autorizzazioni concesse a un'entità in base a diversi tipi di autorizzazioni sono più complesse. Per ulteriori dettagli su come AWS valuta le politiche, consulta. Logica di valutazione delle policy

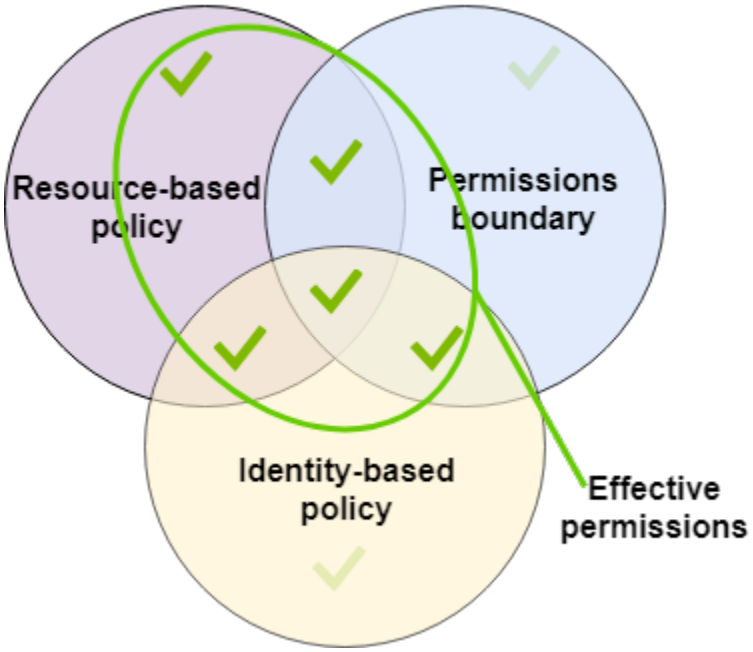
Policy basate su identità con limiti: le policy basate su identità sono policy in linea o gestite collegate a un utente, un gruppo di utenti o un ruolo. Le policy basate su identità concedono autorizzazioni all'entità e i limiti delle autorizzazioni limitano tali autorizzazioni. Le autorizzazioni effettive sono l'intersezione di entrambi i tipi di policy. Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione.



Policy basate su risorse: le policy basate su risorse controllano il modo in cui l'entità principale specificata può accedere alla risorsa a cui la policy è collegata.

Policy basate su risorse per utenti IAM

> All'interno dello stesso account, le politiche basate sulle risorse che concedono autorizzazioni all'ARN di un utente IAM (ovvero, non una sessione come utente federato) non sono limitate da un rifiuto implicito in una policy basata su identità o in un limite delle autorizzazioni.



Policy basate sulle risorse per ruoli IAM

Ruolo IAM: i criteri basati sulle risorse che concedono le autorizzazioni a un ARN del ruolo IAM sono limitati da un rifiuto implicito in un limite delle autorizzazioni o in una policy di sessione.

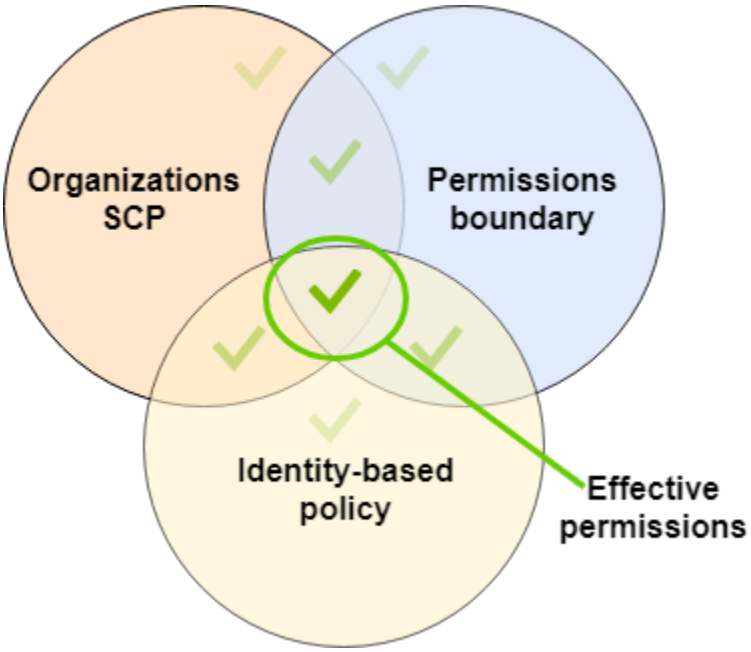
Sessione come ruolo IAM: all'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN della sessione come ruolo IAM concedono le autorizzazioni direttamente alla sessione come ruolo assunto. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione. Quando si assume un ruolo e si effettua una richiesta, il principale che effettua la richiesta è l'ARN della sessione come ruolo IAM e non l'ARN del ruolo stesso.

Policy basate sulle risorse per le sessioni di ruoli IAM e utenti federati

Sessioni come utente federato IAM: una sessione come utente federato IAM è una sessione creata chiamando `GetFederationToken`. Quando un utente federato effettua una richiesta, il principale che effettua la richiesta è l'ARN dell'utente federato e non l'ARN dell'utente IAM che ha eseguito la federazione. All'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN dell'utente federato concedono le autorizzazioni direttamente alla sessione. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione.

Tuttavia, se una policy basata sulle risorse concede l'autorizzazione all'ARN dell'utente IAM che ha eseguito la federazione, le richieste fatte dall'utente federato durante la sessione sono limitate da un rifiuto implicito in un limite di autorizzazione o in una policy di sessione.

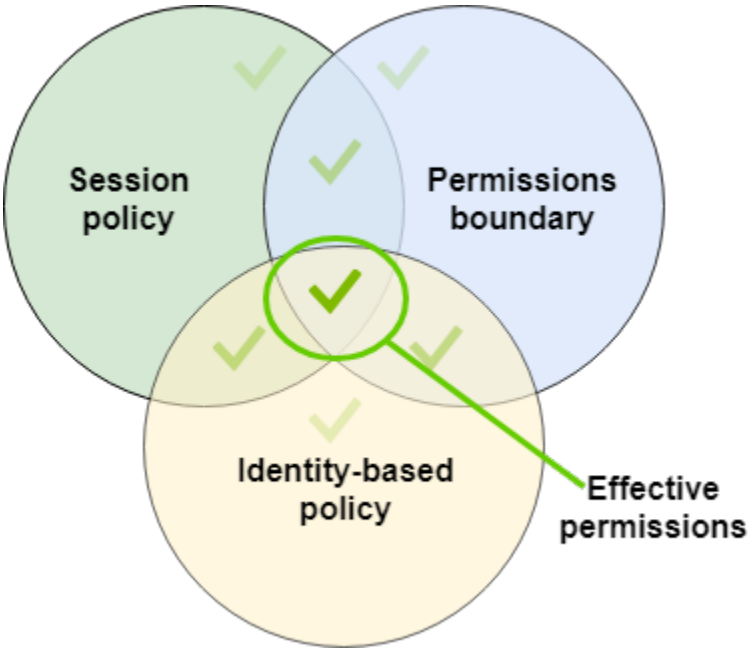
SCP di Organizations: le SCP vengono applicate a un intero Account AWS. Limitano le autorizzazioni per ogni richiesta effettuata da un'entità principale all'interno dell'account. Un'entità IAM (utente o ruolo) può effettuare una richiesta che è influenzata da una SCP, un limite delle autorizzazioni e una policy basata su identità. In questo caso, la richiesta è consentita solo se tutti e tre i tipi di policy la consentono. Le autorizzazioni effettive sono l'intersezione di tutti e tre i tipi di policy. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.



Puoi scoprire se il tuo account è un membro di un'organizzazione in AWS Organizations. I membri dell'organizzazione potrebbero essere influenzati da una SCP. Per visualizzare questi dati utilizzando il AWS CLI comando o l'operazione AWS API, devi disporre delle autorizzazioni per l'`organizations:DescribeOrganization`azione per la tua entità Organizations. È necessario disporre delle autorizzazioni aggiuntive per eseguire l'operazione nella console Organizations. Per sapere se un SCP sta negando l'accesso a una richiesta specifica o per modificare le autorizzazioni effettive, contatta il tuo amministratore. AWS Organizations

Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni per una sessione provengono dall'entità IAM (utente o ruolo) utilizzata per creare la sessione e dalla policy di sessione. Le autorizzazioni della policy basata su identità

dell'entità sono limitate dalla policy di sessione e dal limite delle autorizzazioni. Le autorizzazioni effettive per questo set di tipi di policy sono l'intersezione di tutti e tre i tipi di policy. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sulle policy di sessione, consulta la sezione relativa alle policy di sessione.



Delega di responsabilità ad altri mediante i limiti delle autorizzazioni

Puoi utilizzare il limiti delle autorizzazioni per delegare le attività di gestione delle autorizzazioni, ad esempio la creazione di utenti, agli utenti IAM nel tuo account. Questo consente ad altri di eseguire operazioni a tuo nome all'interno di un limite specifico di autorizzazioni.

Ad esempio, supponiamo che María sia l'amministratore di X-Company. Account AWS María vuole delegare l'attività di creazione di utenti a Zhang. Tuttavia, deve accertarsi che gli utenti creati da Zhang siano conformi alle seguenti regole aziendali:

- Gli utenti non possono utilizzare IAM per creare o gestire utenti, gruppi, ruoli o policy.
- Agli utenti viene rifiutato l'accesso al bucket `logs` di Amazon S3 e all'istanza `i-1234567890abcdef0` di Amazon EC2.
- Gli utenti non possono rimuovere le proprie policy limite.

Per applicare queste regole, María completa le attività seguenti, i cui dettagli sono riportati di seguito:

1. María crea la policy gestita `XCompanyBoundaries` da utilizzare come limite delle autorizzazioni per tutti i nuovi utenti nell'account.

2. María crea la policy gestita `DelegatedUserBoundary` e la assegna come limite delle autorizzazioni per Zhang. Maria prende nota dell'ARN del suo utente amministratore e lo usa nel criterio per impedire a Zhang di accedervi.
3. María crea la policy gestita `DelegatedUserPermissions` e la collega alla policy di autorizzazione per Zhang.
4. María comunica a Zhang le sue nuove responsabilità e limitazioni.

Attività 1: María deve prima creare una policy gestita per definire il limite per i nuovi utenti. María deve consentire a Zhang di concedere agli utenti le policy di autorizzazione necessarie, ma vuole che tali utenti abbiano delle limitazioni. A tale scopo, crea questa policy gestita dal cliente, denominata `XCompanyBoundaries`. Questa policy esegue le seguenti operazioni:

- Consente agli utenti l'accesso completo a diversi servizi
- Consente l'accesso autonomo limitato alla console IAM. Ciò significa che è possibile modificare la password dopo aver effettuato l'accesso alla console. Non è possibile impostare la password iniziale. Per consentire questa operazione, aggiungere l'operazione `"*LoginProfile"` all'istruzione `AllowManageOwnPasswordAndAccessKeys`.
- Rifiuta agli utenti l'accesso al bucket di log Amazon S3 o all'istanza Amazon EC2 `i-1234567890abcdef0`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceBoundaries",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*",
                "dynamodb:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIAMConsoleForCredentials",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswordAndAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::logs",
                "arn:aws:s3:::logs/*"
            ]
        },
        {
            "Sid": "DenyEC2Production",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "arn:aws:ec2:*:*:instance/i-1234567890abcdef0"
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. La `ServiceBoundaries` dichiarazione di questa politica consente l'accesso completo ai servizi AWS specificati. Ciò significa che le operazioni di un nuovo utente in questi servizi sono limitate solo dalle policy di autorizzazione collegate all'utente.
2. La dichiarazione `AllowIAMConsoleForCredentials` consente l'accesso per elencare tutti gli utenti IAM. Questo accesso è necessario per navigare nella pagina Users (Utenti) nella AWS

Management Console. Inoltre, consente di visualizzare i requisiti associati alle password per l'account, necessari per modificare la password.

3. L'istruzione `AllowManageOwnPasswordAndAccessKeys` consente agli utenti di gestire solo le proprie chiavi di accesso a livello di programmazione e le password della console. Questo è importante se Zhang o un altro amministratore concede a un nuovo utente una policy di autorizzazione con accesso IAM completo. In tal caso, l'utente può modificare le proprie autorizzazioni o quelle di altri utenti. Questa istruzione impedisce che ciò si verifichi.
4. L'istruzione `DenyS3Logs` nega esplicitamente l'accesso al bucket `logs`.
5. L'istruzione `DenyEC2Production` nega esplicitamente l'accesso all'istanza `i-1234567890abcdef0`.

Attività 2: María vuole consentire a Zhang di creare tutti gli utenti X-Company, ma solo con il limite delle autorizzazioni `XCompanyBoundaries`. A tale scopo, crea questa policy gestita dal cliente, denominata `DelegatedUserBoundary`. Questa policy definisce il numero massimo di autorizzazioni di cui Zhang può disporre.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateOrChangeOnlyWithBoundary",
            "Effect": "Allow",
            "Action": [
                "iam:AttachUserPolicy",
                "iam:CreateUser",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutUserPolicy"
            ],
            "Resource": "*",
            "Condition": {
               "StringEquals": {
                 "iam:PermissionsBoundary": "arn:aws:iam::123456789012:policy/
XCompanyBoundaries"
                }
            }
        },
        {
            "Sid": "CloudWatchAndOtherIAMTasks",
```

```
            "Effect": "Allow",
            "Action": [
              "cloudwatch:*",
              "iam:CreateAccessKey",
              "iam:CreateGroup",
              "iam:CreateLoginProfile",
              "iam:CreatePolicy",
              "iam:DeleteGroup",
              "iam:DeletePolicy",
              "iam:DeletePolicyVersion",
              "iam:DeleteUser",
              "iam:GetAccountPasswordPolicy",
              "iam:GetGroup",
              "iam:GetLoginProfile",
              "iam:GetPolicy",
              "iam:GetPolicyVersion",
              "iam:GetRolePolicy",
              "iam:GetUser",
              "iam:GetUserPolicy",
              "iam:ListAccessKeys",
              "iam:ListAttachedRolePolicies",
              "iam:ListAttachedUserPolicies",
              "iam:ListEntitiesForPolicy",
              "iam:ListGroups",
              "iam:ListGroupsForUser",
              "iam:ListMFADevices",
              "iam:ListPolicies",
              "iam:ListPolicyVersions",
              "iam:ListRolePolicies",
              "iam:ListSSHPublicKeys",
              "iam:ListServiceSpecificCredentials",
              "iam:ListSigningCertificates",
              "iam:ListUserPolicies",
              "iam:ListUsers",
              "iam:SetDefaultPolicyVersion",
              "iam:SimulateCustomPolicy",
              "iam:SimulatePrincipalPolicy",
              "iam:UpdateGroup",
              "iam:UpdateLoginProfile",
              "iam:UpdateUser"
            ],
            "NotResource": "arn:aws:iam::123456789012:user/Maria"
        },
        {
```

```
            "Sid": "NoBoundaryPolicyEdit",
            "Effect": "Deny",
            "Action": [
                "iam:CreatePolicyVersion",
                "iam:DeletePolicy",
                "iam:DeletePolicyVersion",
                "iam:SetDefaultPolicyVersion"
            ],
            "Resource": [
                "arn:aws:iam::123456789012:policy/XCompanyBoundaries",
                "arn:aws:iam::123456789012:policy/DelegatedUserBoundary"
            ]
        },
        {
            "Sid": "NoBoundaryUserDelete",
            "Effect": "Deny",
            "Action": "iam:DeleteUserPermissionsBoundary",
            "Resource": "*"
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. L'istruzione `CreateOrChangeOnlyWithBoundary` consente a Zhang di creare utenti IAM ma solo se utilizza la policy `XCompanyBoundaries` per impostare il limite delle autorizzazioni. L'istruzione gli consente inoltre di impostare il limite delle autorizzazioni per gli utenti esistenti, ma solo utilizzando la stessa policy. Infine, consente a Zhang di gestire le policy di autorizzazione per gli utenti per i quali è stato impostato questo limite delle autorizzazioni.
2. L'istruzione `CloudWatchAndOtherIAMTasks` consente a Zhang di completare altre attività di gestione di utenti, gruppi e policy. Ha le autorizzazioni per reimpostare le password e creare chiavi di accesso per qualsiasi utente IAM non elencato nell'elemento della policy `NotResource`. Questo gli consente di aiutare gli utenti con problemi di accesso.
3. L'istruzione `NoBoundaryPolicyEdit` nega a Zhang l'accesso per aggiornare la policy `XCompanyBoundaries`. Zhang non può modificare alcuna policy utilizzata per impostare il limite delle autorizzazioni per sé o per altri utenti.
4. L'istruzione `NoBoundaryUserDelete` nega a Zhang l'accesso per eliminare il limite delle autorizzazioni per sé o per altri utenti.

María assegna quindi la policy `DelegatedUserBoundary` come limite delle autorizzazioni per l'utente `Zhang`.

Attività 3: poiché il limite delle autorizzazioni controlla il numero massimo di autorizzazioni, ma non concede l'accesso di per sé, María deve creare una policy di autorizzazione per Zhang. A tale scopo, crea questa policy, denominata `DelegatedUserPermissions`. Questa policy definisce le operazioni che Zhang può eseguire, entro il limite definito.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": "iam:*",
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLimited",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetDashboard",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListDashboards",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3BucketContents",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::ZhangBucket"
        }
    ]
}
```

Ogni istruzione svolge una funzione diversa:

1. L'istruzione `IAM` della policy consente a Zhang l'accesso completo a IAM. Tuttavia, poiché il limite delle autorizzazioni di Zhang consente solo alcune operazioni in IAM, le sue autorizzazioni valide in IAM sono limitate solo dal relativo limite delle autorizzazioni.
2. La `CloudWatchLimited` dichiarazione consente a Zhang di eseguire cinque azioni in CloudWatch. Il suo limite di autorizzazioni consente l'accesso a tutte le azioni CloudWatch, quindi le sue CloudWatch autorizzazioni effettive sono limitate solo dalla sua politica sulle autorizzazioni.
3. L'istruzione `S3BucketContents` consente a Zhang di visualizzare il bucket `ZhangBucket` di Amazon S3. Tuttavia, il limite delle autorizzazioni di Zhang non gli consente alcuna operazione in Amazon S3, quindi non può eseguire operazioni S3, indipendentemente dalla sua policy di autorizzazione.

> **Note**
>
> Le policy di Zhang gli permettono di creare un utente in grado di accedere alle risorse Amazon S3 a cui lui non può accedere. Delegando queste operazioni amministrative, Maria di fatto si fida dell'accesso di Zhang ad Amazon S3.

María collega quindi la policy `DelegatedUserPermissions` come policy di autorizzazione per l'utente `Zhang`.

Attività 4: María fornisce a Zhang le istruzioni per creare un nuovo utente. Zhang può creare nuovi utenti con tutte le autorizzazioni necessarie, ma deve assegnare loro la policy `XCompanyBoundaries` come limite delle autorizzazioni.

Zhang completa le attività seguenti:

1. Zhang crea un utente mediante la AWS Management Console. Digita il nome utente `Nikhil` e consente l'accesso alla console a tale utente. Cancella la casella di controllo accanto a Richiede reimpostazione della password poiché le policy sopra riportate consentono agli utenti di modificare la password solo dopo aver effettuato l'accesso alla console IAM.
2. Nella pagina Imposta le autorizzazioni, Zhang sceglie le politiche di autorizzazione IAM FullAccess e AmazonS3 ReadOnlyAccess che consentono a Nikhil di svolgere il suo lavoro.
3. Zhang salta la sezione Set permissions boundary (Imposta limite delle autorizzazioni), dimenticando le indicazioni di María.
4. Zhang esamina i dettagli utente e seleziona Create user (Crea utente).

L'operazione ha esito negativo e l'accesso viene negato. In base al limite delle autorizzazioni di Zhang, `DelegatedUserBoundary`, qualsiasi utente da lui creato deve includere la policy `XCompanyBoundaries` come limite delle autorizzazioni.

5. Zhang torna alla pagina precedente. Nella sezione Set permissions boundary (Imposta limite delle autorizzazioni), seleziona la policy `XCompanyBoundaries`.
6. Zhang esamina i dettagli utente e seleziona Create user (Crea utente).

   L'utente viene creato.

Quando Nikhil esegue l'accesso, può accedere a IAM e Amazon S3, ma non alle operazioni rifiutate dal suo limite delle autorizzazioni. Ad esempio, può modificare la propria password in IAM ma non può creare un altro utente o modificare le policy. Nikhil ha accesso in sola lettura ad Amazon S3.

Se qualcuno aggiunge una policy basata sulle risorse al bucket `logs` che consente a Nikhil di inserire un oggetto nel bucket, significa che non può ancora accedere al bucket. Questo perché qualsiasi operazione sul bucket `logs` è esplicitamente rifiutata dal suo limite delle autorizzazioni. Un rifiuto esplicito in qualsiasi tipo di policy determina il rifiuto di una richiesta. Tuttavia, se una policy basata sulle risorse collegata a un segreto di Secrets Manager consente a Nikhil di eseguire l'operazione `secretsmanager:GetSecretValue`, allora Nikhil potrà recuperare e decrittare il segreto. Questo perché le operazioni di Secrets Manager non sono esplicitamente rifiutate dal suo limite delle autorizzazioni e i rifiuti impliciti nei limiti delle autorizzazioni non limitano le policy basate sulle risorse.

Policy basate sulle identità e policy basate su risorse

Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. Quando crei una policy di autorizzazione per limitare l'accesso a una risorsa, puoi scegliere una policy basata su identità o una policy basata su risorse.

Le policy basate su identità sono collegate a un utente, un gruppo o un ruolo IAM. Queste policy consentono di specificare cosa può fare quell'identità (le sue autorizzazioni). Ad esempio, puoi collegare la policy all'utente IAM di nome John, dichiarando che a questo utente è autorizzato ad eseguire l'operazione `RunInstances` di Amazon EC2. La policy potrebbe inoltre dichiarare che a John è consentito ottenere oggetti da una tabella di Amazon DynamoDB denominata `MyCompany`. È inoltre possibile consentire a John di gestire le proprie credenziali di sicurezza IAM. Le policy basate su identità sulle identità possono essere gestite o inline.

Le policy basate su risorse sono collegate a una risorsa. Ad esempio, puoi collegare policy basate sulle risorse a bucket Amazon S3, code Amazon SQS, endpoint AWS Key Management Service VPC, chiavi di crittografia e tabelle e flussi Amazon DynamoDB. Per un elenco dei servizi che supportano le policy basate su risorse, consulta AWS servizi che funzionano con IAM.

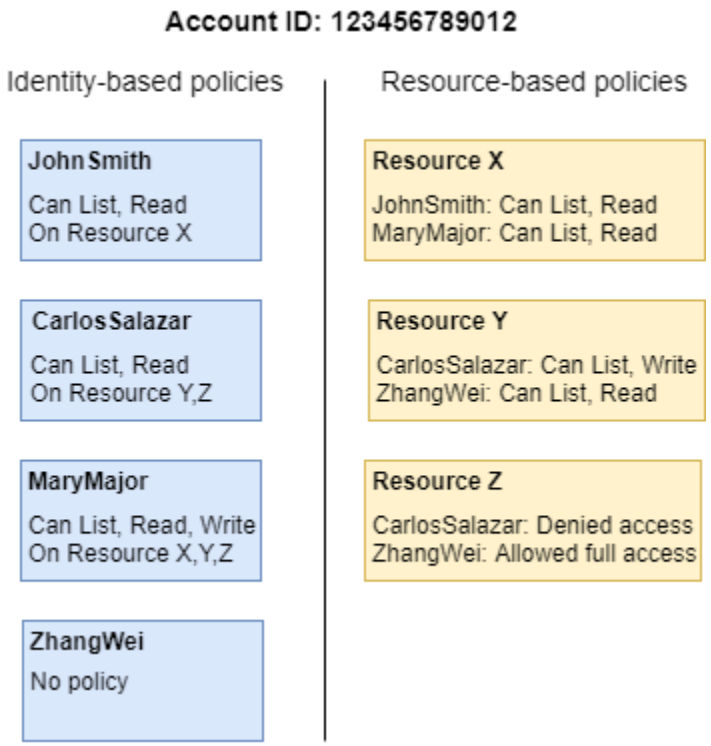
Le policy basate su risorse consentono di specificare quali utenti hanno accesso a una risorsa e quali operazioni possono eseguirvi. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?. Le policy basate su risorse sono solo inline, non gestite.

> **Note**
>
> Le policy basate su risorse sono diverse dalle autorizzazioni a livello di risorsa. È possibile collegare policy basate su risorse direttamente a una risorsa, come descritto in questo argomento. Le autorizzazioni a livello di risorsa si riferiscono alla possibilità di utilizzare gli ARN per specificare le singole risorse in una policy. Le AWS politiche basate sulle risorse sono supportate solo da alcuni servizi. Per un elenco dei servizi che supportano delle policy basate su risorse e le autorizzazioni a livello di risorsa, consultare AWS servizi che funzionano con IAM.

Per informazioni su come interagiscono le policy basate su identità e le policy basate sulle risorse all'interno dello stesso account, consulta Valutazione delle policy in un singolo account.

Per informazioni su come le policy interagiscono tra gli account, consulta Logica di valutazione della policy multiaccount.

Per comprendere meglio questi concetti, visualizza la figura riportata di seguito. L'amministratore dell'account `123456789012` ha collegato policy basate su identità agli utenti `JohnSmith`, `CarlosSalazar` e `MaryMajor`. Alcune delle operazioni in queste policy possono essere eseguite su risorse specifiche. Ad esempio, l'utente `JohnSmith` può eseguire alcune operazioni su `Resource X`. Si tratta di un'autorizzazione a livello di risorsa in una policy basata su identità. L'amministratore inoltre ha aggiunto policy basate su risorse a `Resource X`, `Resource Y` e `Resource Z`. Le policy basate su risorse consentono di specificare chi può accedere alla risorsa. Ad esempio, la policy basata su risorsa su `Resource X` consente agli utenti `JohnSmith` e `MaryMajor` l'accesso all'elenco e in lettura a quella risorsa.



L'esempio di account `123456789012` consente agli utenti seguenti di eseguire le operazioni elencate:

- JohnSmith— John può eseguire azioni su elenchi e leggere. `Resource X` Gli è concessa questa autorizzazione dalla policy basata su identità sul suo utente e dalla policy basata su risorsa su `Resource X`.
- CarlosSalazar— Carlos può eseguire azioni di elenco, lettura e scritturaResource Y, ma gli viene negato l'accesso aResource Z. La policy basata su identità su Carlos gli consente di eseguire operazioni di lettura ed elenco su `Resource Y`. La policy basata sulla risorsa `Resource Y`, inoltre, gli concede autorizzazioni di scrittura. Tuttavia, anche se la sua policy basata su identità gli consente l'accesso a `Resource Z`, la policy basata sulla risorsa `Resource Z` nega tale accesso. Un `Deny` esplicito sostituisce un `Allow` e il suo accesso a `Resource Z` viene negato. Per ulteriori informazioni, consulta Logica di valutazione delle policy.
- MaryMajor— Mary può eseguire operazioni di elenco, lettura e scrittura su `Resource X`Resource Y, e`Resource Z`. La sua policy basata su identità le consente più operazioni su più risorse rispetto alle policy basate su risorse, ma nessuna di queste nega l'accesso.
- ZhangWei— Zhang ha pieno accesso a. `Resource Z` Zhang non ha policy basate su identità, ma la policy basata sulla risorsa `Resource Z` gli consente l'accesso completo alla risorsa. Zhang può anche eseguire elenco e leggere operazioni su `Resource Y`.

Le policy basate su identità e le policy basate su risorse sono entrambe policy di autorizzazione e vengono valutate insieme. Per una richiesta a cui si applicano solo le politiche di autorizzazione, verifica AWS innanzitutto che tutte le politiche contengano un. `Deny` Se ne esiste una, la richiesta viene negata. Quindi, AWS verifica ogni `Allow`. Se almeno un'istruzione della policy consente l'operazione nella richiesta, la richiesta è consentita. Non importa se l'`Allow` è concessa dalla policy basata su identità o dalla policy basata su risorse.

> **Important**
>
> Questa logica si applica solo quando la richiesta viene effettuata all'interno di un singolo Account AWS. Per le richieste effettuate da un account a un altro, il richiedente nell'`Account A` deve disporre di una policy basata su identità che gli consenta di effettuare una richiesta alla risorsa nell'`Account B`. Inoltre, la policy basata sulla risorsa nell'`Account B` deve consentire al richiedente nell'`Account A` di accedere alla risorsa. In entrambi gli account devono essere presenti policy che consentono l'operazione, altrimenti la richiesta non riesce. Per ulteriori informazioni sull'utilizzo delle policy basate sulle risorse per l'accesso tra account, consulta Accesso alle risorse multi-account in IAM.

Un utente che dispone di autorizzazioni specifiche potrebbe richiedere una risorsa a cui è collegata anche una policy di autorizzazione. In tal caso, AWS valuta entrambi i set di autorizzazioni per determinare se concedere l'accesso alla risorsa. Per ulteriori informazioni su come vengono valutate le policy, consultare Logica di valutazione delle policy.

> **Note**
>
> Amazon S3 supporta le policy basate sulle identità e le policy basate su risorse (dette policy dei bucket). Inoltre, Amazon S3 supporta un meccanismo di autorizzazione noto come lista di controllo accessi (ACL) che è indipendente dalle policy e dalle autorizzazioni IAM. È possibile usare le policy IAM in combinazione con le liste di controllo accessi di Amazon S3. Per ulteriori informazioni, consulta Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service.

Controllo dell'accesso alle AWS risorse tramite policy

Puoi utilizzare una policy per controllare l'accesso alle risorse all'interno di IAM o di tutte AWS.

Per utilizzare una policy per controllare l'accesso in AWS, è necessario comprendere in che modo AWS concede l'accesso. AWS è composto da raccolte di risorse. Un utente IAM è una risorsa. Un bucket Amazon S3 è una risorsa. Quando si utilizza l' AWS API AWS CLI, il o il AWS Management Console per eseguire un'operazione (come la creazione di un utente), si invia una richiesta per tale operazione. La richiesta specifica un'operazione, una risorsa, un'entità principale (utente o ruolo), un account principale e qualsiasi altra informazione necessaria per la richiesta. Tutte queste informazioni forniscono il contesto.

AWS verifica quindi che tu (il principale) sia autenticato (effettuato l'accesso) e autorizzato (disponi del permesso) a eseguire l'azione specificata sulla risorsa specificata. Durante l'autorizzazione, AWS controlla tutte le politiche che si applicano al contesto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON e specifica le autorizzazioni per le entità principali. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Policy e autorizzazioni in IAM.

AWS autorizza la richiesta solo se ogni parte della richiesta è consentita dalle politiche. Per visualizzare un diagramma di questo processo, consultare Funzionamento di IAM. Per informazioni dettagliate su come AWS determinare se una richiesta è consentita, consultaLogica di valutazione delle policy.

Quando crei una policy IAM, puoi controllare l'accesso ai seguenti elementi:

- Principali: controlla quello che la persona che effettua la richiesta (principale) è autorizzata a fare.
- Identità IAM: controlla a quali identità IAM (gruppi di utenti, utenti e ruoli) è possibile accedere e come.
- Policy IAM: controlla quali utenti possono creare, modificare ed eliminare le policy gestite dai clienti e quali utenti possono collegare e scollegare tutte le policy gestite.
- Risorse AWS: consente di controllare quali utenti hanno accesso alle risorse tramite una policy basata sulle identità o una policy basata sulle risorse.
- Account AWS: consente di controllare se una richiesta è consentita solo per i membri di un determinato account.

Le politiche consentono di specificare chi ha accesso alle AWS risorse e quali azioni può eseguire su tali risorse. Inizialmente, nessun utente IAM dispone di autorizzazioni. Ovvero, per impostazione predefinita, gli utenti non possono eseguire alcuna operazione, neppure visualizzare le proprie chiavi di accesso. Per fornire a un utente l'autorizzazione per eseguire un'operazione, è possibile aggiungere l'autorizzazione all'utente, ovvero collegare una policy all'utente. In alternativa, puoi aggiungere l'utente a un gruppo di utenti con l'autorizzazione desiderata.

Ad esempio, è possibile concedere a un utente l'autorizzazione per l'elencazione delle proprie chiavi di accesso. È inoltre possibile espandere tale autorizzazione e consentire inoltre a ciascun utente di creare, aggiornare ed eliminare le proprie chiavi.

Quando concedi le autorizzazioni a un gruppo di utenti, tutti gli utenti in quel gruppo potranno usufruire di tali autorizzazioni. Ad esempio, puoi concedere al gruppo di utenti Administrators l'autorizzazione a eseguire qualsiasi azione IAM su qualsiasi Account AWS risorsa. Un altro esempio: è possibile fornire al gruppo di utenti Manager l'autorizzazione per descrivere le istanze Amazon EC2 dell' Account AWS.

Per informazioni su come delegare le autorizzazioni di base per utenti, gruppi di utenti e ruoli, consulta Autorizzazioni necessarie per accedere alle risorse IAM. Per ulteriori esempi di policy che utilizzano queste autorizzazioni, consultare Esempi di policy per amministrare le risorse IAM.

Controllo dell'accesso per le entità principali

È possibile usare le policy per controllare le operazioni che la persona da cui proviene la richiesta (entità principale) è autorizzata a effettuare. A tale scopo, è necessario collegare una policy basata su identità all'identità di questa persona (utente, gruppo di utenti o ruolo). È possibile utilizzare anche un limite di autorizzazioni per impostare il numero massimo di autorizzazioni che un'entità (utente o ruolo) può avere.

Ad esempio, supponiamo di volere che l'utente Zhang Wei abbia accesso completo ad Amazon DynamoDB CloudWatch, Amazon EC2 e Amazon S3. È possibile creare due policy diverse, in modo che successivamente sia possibile suddividerle nel caso sia necessario un set di autorizzazioni per un utente diverso. In alternativa, è possibile includere entrambe le autorizzazioni in una singola policy e quindi collegare tale policy all'utente IAM denominato Zhang Wei. È anche possibile collegare una policy a un gruppo di utenti a cui Zhang appartiene o a un ruolo che Zhang può assumere. Di conseguenza, quando Zhang visualizza i contenuti di un bucket di S3, le sue richieste vengono accettate. Se prova a creare un nuovo utente IAM, la richiesta viene rifiutata perché non dispone dell'autorizzazione.

È possibile utilizzare un limite delle autorizzazioni su Zhang per fare in modo che non gli venga mai dato accesso al bucket `DOC-EXAMPLE-BUCKET1` di S3. A tale scopo, è necessario determinare il numero massimo di autorizzazioni per Zhang. In questo caso è possibile controllare le sue attività con le policy di autorizzazione. L'importante è che non possa accedere al bucket riservato. Quindi usi la seguente policy per definire il limite di Zhang per consentire tutte le AWS azioni per Amazon S3 e alcuni altri servizi ma negare l'accesso al bucket S3. `DOC-EXAMPLE-BUCKET1` Poiché il limite delle

autorizzazioni non consente alcuna operazione IAM, impedisce a Zhang di eliminare il proprio limite o quello di altri utenti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PermissionsBoundarySomeServices",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*",
                "dynamodb:*",
                "ec2:*",
                "s3:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionsBoundaryNoConfidentialBucket",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
            ]
        }
    ]
}
```

Quando si assegna una policy come questa come limite delle autorizzazioni per un utente, la policy non concede alcuna autorizzazione. Imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM. Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.

Per informazioni dettagliate sulle procedure precedenti, è possibile consultare le risorse seguenti:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Creazione di policy IAM.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

- Per consultare un esempio di policy per concedere accesso completo a EC2, consultare Amazon EC2: consente l'accesso completo a EC2 entro una regione specifica, a livello di programmazione e nella console.
- Per permettere l'accesso in sola lettura a un bucket S3, utilizzare le prime due istruzioni della seguente policy di esempio: Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console.
- Per visualizzare una policy di esempio che consente agli utenti di impostare le credenziali, ad esempio la password della console, le chiavi di accesso a livello di programmazione e i dispositivi MFA, consulta la pagina AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Controllo dell'accesso alle identità

Puoi usare le policy IAM per controllare le operazioni che tutti gli utenti possono eseguire per un'identità mediante la creazione di una policy da collegare a tutti gli utenti tramite un gruppo di utenti. Per eseguire questa operazione, è necessario creare una policy che limita le operazioni che possono essere eseguite per un'identità oppure gli utenti che possono accedere.

Ad esempio, puoi creare un gruppo di utenti denominato AllUserse quindi associare quel gruppo di utenti a tutti gli utenti. Quando si crea il gruppo di utenti, è possibile fornire a tutti gli utenti l'accesso per impostare le proprie credenziali come descritto nella sezione precedente. È quindi possibile creare una policy che rifiuti l'accesso alla modifica del gruppo di utenti a meno che il nome utente non sia incluso nella condizione della policy. Tuttavia, la parte della policy rifiuta solo l'accesso a chiunque eccetto gli utenti elencati. È inoltre necessario includere le autorizzazioni per permettere tutte le operazioni di gestione del gruppo di utenti per tutti gli utenti del gruppo. Infine, puoi collegare questa policy al gruppo di utenti in modo che venga applicata a tutti gli utenti. Di conseguenza, quando un utente non specificato nella policy cerca di apportare modifiche al gruppo di utenti, la richiesta viene rifiutata.

Per creare questa policy con l'editor visivo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione Visivo.
5. In Seleziona un servizio, scegli IAM.
6. In Operazioni consentite, digita **`group`** nella casella di ricerca. L'editor visivo mostra tutte le operazioni IAM che contengono la parola `group`. Selezionare tutte le caselle di controllo.
7. Selezionare Resources (Risorse) per specificare le risorse per la policy. In base alle operazioni scelte, dovrebbero venire visualizzati i tipi di risorse gruppo e utente.

   - gruppo: scegli Aggiungi ARN. Per Risorse in, seleziona l'opzione Qualsiasi account. Seleziona la casella di controllo Qualsiasi nome di gruppo con percorso, quindi digita il nome del gruppo di utenti **`AllUsers`**. Quindi scegli Aggiungi ARN.
   - utente: seleziona la casella di controllo accanto a Qualsiasi in questo account.

   Una delle operazioni scelte, `ListGroups`, non supporta l'utilizzo di risorse specifiche. Non è necessario selezionare All resources (Tutte le risorse) per tale operazione. Quando salvi la policy o la visualizzi nell'editor JSON, puoi notare che IAM crea automaticamente un nuovo blocco di autorizzazioni che concede l'autorizzazione per questa operazione a tutte le risorse.
8. Per aggiungere un altro blocco di autorizzazioni, scegli Aggiungi altre autorizzazioni.
9. Scegli Seleziona un servizio e quindi IAM.
10. Scegli Operazioni consentite, quindi seleziona Passa ad autorizzazioni rifiutate. In questo caso, l'intero blocco viene utilizzato per rifiutare le autorizzazioni.
11. Nella casella di ricerca digitare **`group`**. L'editor visivo mostra tutte le operazioni IAM che contengono la parola `group`. Selezionare le caselle di controllo accanto alle seguenti operazioni:

    - CreateGroup
    - DeleteGroup
    - RemoveUserFromGroup
    - AttachGroupPolicy
    - DeleteGroupPolicy
    - DetachGroupPolicy
    - PutGroupPolicy
    - UpdateGroup

12. Selezionare Resources (Risorse) per specificare le risorse per la policy. In base alle operazioni scelte, dovrebbe venire visualizzato il tipo di risorse group (gruppo). Scegli Aggiungi ARN. Per Risorse in, seleziona l'opzione Qualsiasi account. Per Qualsiasi nome gruppo con percorso, digita il nome del gruppo di utenti **`AllUsers`**. Quindi scegli Aggiungi ARN.
13. Scegli Condizioni di richiesta - opzionale, quindi scegli Aggiungi altra condizione. Completare il modulo con i seguenti valori:

    - Chiave di condizione: scegli aws:username
    - Qualificatore: scegli Default
    - Operatore: scegli StringNotEquals
    - Valore: digita **`srodriguez`**, quindi scegli Aggiungi per aggiungere un altro valore. Digita **`mjackson`**, quindi scegli Aggiungi per aggiungere un altro valore. Digita **`adesai`** e quindi seleziona Aggiungi un altro valore di condizione.

    Questa condizione garantisce che l'accesso venga rifiutato per le operazioni di gestione del gruppo di utenti specificate se l'utente che effettua la chiamata non è incluso nell'elenco. Poiché l'autorizzazione viene rifiutata in modo esplicito, il blocco precedente che consentiva a tali utenti di chiamare le operazioni viene ignorato. Per gli utenti inclusi nell'elenco, l'accesso non viene rifiutato e viene concessa dell'autorizzazione del primo blocco di autorizzazioni, in modo che possano gestire completamente il gruppo.
14. Quando hai terminato, seleziona Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'opzione dell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

15. Nella pagina Verifica e crea, digita **`LimitAllUserGroupManagement`** in Nome della policy. In Description (Descrizione), digitare **`Allows all users read-only access to a specific user group, and allows only specific users access to make changes to the user group`**. Rivedi il campo Autorizzazioni definite in questa policy per accertarti di disporre delle autorizzazioni previste. Quindi selezionare Create policy (Crea policy) per salvare la nuova policy.

16. Collega la policy al tuo gruppo di utenti. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

In alternativa, è possibile creare la stessa policy utilizzando questo esempio di documento di policy JSON. Per visualizzare questa policy JSON, consulta IAM: consente a utenti IAM specifici di gestire un gruppo a livello di programmazione e nella console. Per istruzioni dettagliate sulla creazione di una policy utilizzando un documento JSON, consulta the section called “Creazione di policy utilizzando l'editor JSON”.

Controllo dell'accesso alle policy

Puoi controllare in che modo gli utenti possono applicare le policy AWS gestite. Per eseguire questa operazione, collegare questa policy per tutti gli utenti. In teoria, è possibile eseguire questa operazione utilizzando un gruppo di utenti.

Ad esempio, potresti creare una policy che consenta agli utenti di collegare solo le policy IAM UserChangePassword e PowerUserAccess AWS gestite a un nuovo utente, gruppo di utenti o ruolo IAM.

Per le policy gestite dal cliente, è possibile controllare chi può creare, aggiornare ed eliminare queste policy. È possibile controllare chi può collegare e distaccare le policy per le entità principali (utenti, gruppi di utenti e ruoli). È inoltre possibile controllare quali policy un utente può collegare o distaccare e per quale entità.

Ad esempio, è possibile concedere autorizzazioni a un account amministratore per creare, aggiornare ed eliminare le policy. Quindi è possibile assegnare le autorizzazioni a un team leader o a un altro amministratore limitato collegare o distaccare queste policy a/da entità principali che l'amministratore limitato gestisce.

Per ulteriori informazioni, fare riferimento a queste risorse:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Creazione di policy IAM.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
- Per consultare un esempio di policy per limitare l'uso di policy gestito, consultare IAM: limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM..

Controllo delle autorizzazioni per la creazione, l'aggiornamento e l'eliminazione di policy gestite dal cliente

Puoi usare le policy IAM per controllare chi può creare, aggiornare ed eliminare le policy gestite dal cliente nell' Account AWS. L'elenco seguente contiene le operazioni di API che si riferiscono direttamente a creazione, aggiornamento ed eliminazione di policy o versioni di policy:

- CreatePolicy
- CreatePolicyVersion
- DeletePolicy
- DeletePolicyVersion
- SetDefaultPolicyVersion

Le operazioni API nell'elenco precedente corrispondono alle operazioni che è possibile consentire o rifiutare, ovvero le autorizzazioni che è possibile concedere, utilizzando una policy IAM.

Esaminiamo l'esempio di policy seguente. Permette a un utente di creare, aggiornare (ovvero creare una nuova versione della policy), eliminare e impostare una versione predefinita per tutte le policy gestite dal cliente nell' Account AWS. La policy di esempio, inoltre, consente all'utente di elencare e ottenere le policy. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

Example Esempio di policy che permette la creazione, l'aggiornamento, l'eliminazione, la visualizzazione, l'ottenimento e la configurazione della versione di default per tutte le policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "iam:CreatePolicy",
      "iam:CreatePolicyVersion",
      "iam:DeletePolicy",
      "iam:DeletePolicyVersion",
      "iam:GetPolicy",
      "iam:GetPolicyVersion",
      "iam:ListPolicies",
      "iam:ListPolicyVersions",
```

```
      "iam:SetDefaultPolicyVersion"
    ],
    "Resource": "*"
  }
}
```

È possibile creare policy che limitano l'uso di queste operazioni delle API che interessano solo le policy gestite specificate dall'utente. Ad esempio, è possibile permettere a un utente di impostare la versione predefinita ed eliminare le versioni di policy, ma solo per determinate policy gestite dal cliente. A tale scopo, è necessario specificare l'ARN della policy nell'elemento `Resource` della policy che concede l'autorizzazione.

L'esempio seguente mostra una policy che consente a un utente di eliminare versioni della policy e impostare la versione predefinita. Tuttavia, queste azioni sono consentite solo per le policy gestite dal cliente che includono il percorso /TEAM-A/. L'ARN della policy gestita dal cliente è specificato nell'elemento `Resource` della policy. In questo esempio l'ARN include un percorso e un carattere jolly e quindi corrisponde a tutte le policy gestite dal cliente che includono il percorso /TEAM-A/. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

Per ulteriori informazioni sull'utilizzo di percorsi di clienti nei nomi delle policy gestite dal cliente, consultare Nomi descrittivi e percorsi.

Example Esempio di policy che consente di eliminare le versioni di policy e impostare la versione di default solo per policy specifiche

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "iam:DeletePolicyVersion",
      "iam:SetDefaultPolicyVersion"
    ],
    "Resource": "arn:aws:iam::account-id:policy/TEAM-A/*"
  }
}
```

Controllo delle autorizzazioni per collegare e distaccare le policy gestite

È possibile utilizzare le policy IAM anche per permettere agli utenti di utilizzare policy gestite specifiche. In pratica, è possibile controllare le autorizzazioni che un utente può concedere ad altre entità principali.

L'elenco seguente mostra le operazioni di API che consentono direttamente di collegare e distaccare le policy gestite a/da entità principali:

- AttachGroupPolicy
- AttachRolePolicy
- AttachUserPolicy
- DetachGroupPolicy
- DetachRolePolicy
- DetachUserPolicy

È possibile creare policy che limitano l'uso di queste operazioni delle API che interessano solo policy gestite specifiche e/o entità del principale specificate dall'utente. Ad esempio, è possibile permettere a un utente di collegare le policy gestite, ma per le policy specificate dall'utente. Oppure, è possibile permettere a un utente di collegare le policy gestite, ma alle entità del principale specificate dall'utente.

L'esempio di policy seguente consente a un utente di collegare le policy gestite solo ai gruppi di utenti e ai ruoli che includono il percorso /TEAM-A/. Gli ARN del gruppo di utenti e del ruolo sono specificati nell'elemento `Resource` della policy. In questo esempio gli ARN includono un percorso e un carattere jolly e quindi corrispondono a tutti i gruppi di utenti e ai ruoli che includono il percorso /TEAM-A/. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

Example Esempio di policy che consente di collegare policy gestite solo a gruppi di utenti o ruoli specifici

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
```

```
    "Action": [
      "iam:AttachGroupPolicy",
      "iam:AttachRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::account-id:group/TEAM-A/*",
      "arn:aws:iam::account-id:role/TEAM-A/*"
    ]
  }
}
```

È possibile limitare ulteriormente le operazioni dell'esempio precedente per influire solo su determinate policy. In altre parole, puoi controllare le autorizzazioni che un utente può collegare ad altre entità principali, aggiungendo una condizione alla policy.

In questo esempio, la condizione garantisce che le autorizzazioni `AttachGroupPolicy` e `AttachRolePolicy` siano consentite solo quando la policy da collegare corrisponde a uno delle policy specificate. La condizione utilizza la chiave della condizione `iam:PolicyARN` per determinare quali policy possono essere collegate. L'esempio di policy seguente amplia il concetto espresso nell'esempio precedente. Consente a un utente di collegare solo le policy gestite che includono il percorso /TEAM-A/ solo ai gruppi di utenti e ai ruoli che includono il percorso /TEAM-A/. Per ulteriori informazioni su come creare una policy utilizzando questo esempio di documento di policy JSON, consultare the section called “Creazione di policy utilizzando l'editor JSON”.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "iam:AttachGroupPolicy",
      "iam:AttachRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::account-id:group/TEAM-A/*",
      "arn:aws:iam::account-id:role/TEAM-A/*"
    ],
    "Condition": {"ArnLike":
      {"iam:PolicyARN": "arn:aws:iam::account-id:policy/TEAM-A/*"}
    }
  }
}
```

Questa policy utilizza l'operatore di condizione `ArnLike` ma puoi anche utilizzare l'operatore di condizione `ArnEquals` perché questi due operatori si comportano in modo identico. Per ulteriori informazioni su `ArnLike` e `ArnEquals`, consulta Operatori di condizione con Amazon Resource Name (ARN) nella sezione Tipi di condizione dei Riferimenti agli elementi della policy.

Ad esempio, è possibile limitare l'uso di operazioni per coinvolgere solo le policy gestite specificate dall'utente. A tale scopo, è necessario specificare l'ARN della policy nell'elemento `Condition` della policy che concede l'autorizzazione. Ad esempio, per specificare l'ARN di una policy gestita dal cliente:

```
"Condition": {"ArnEquals":
  {"iam:PolicyARN": "arn:aws:iam::123456789012:policy/POLICY-NAME"}
}
```

È inoltre possibile specificare l'ARN di una policy AWS gestita nell'elemento di `Condition` una policy. L'ARN di una policy AWS gestita utilizza l'alias speciale `aws` nella policy ARN anziché un ID account, come in questo esempio:

```
"Condition": {"ArnEquals":
  {"iam:PolicyARN": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"}
}
```

Controllo dell'accesso alle risorse

È possibile controllare chi ha accesso alle risorse utilizzando una policy basata sulle identità o una policy basata sulle risorse. In una policy basata sulle identità si collega la policy a un'identità e si specifica a quali risorse può accedere tale identità. In una policy basata sulle risorse, si collega una policy alla risorsa che si desidera controllare. Nella policy, è necessario specificare quali entità principali possono accedere a tale risorsa. Per ulteriori informazioni su entrambi questi tipi di policy, consultare Policy basate sulle identità e policy basate su risorse.

Per ulteriori informazioni, fare riferimento a queste risorse:

- Per ulteriori informazioni sulla creazione di una policy IAM che è possibile collegare a un principale, consulta Creazione di policy IAM.
- Per ulteriori informazioni su come collegare una policy IAM a un principale, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
- Amazon S3 supporta l'utilizzo delle policy basate su risorse nei relativi bucket. Per ulteriori informazioni, consulta Esempi di policy di bucket.

Gli autori delle risorse non dispongono automaticamente di autorizzazioni

Se accedi utilizzando le Utente root dell'account AWS credenziali, sei autorizzato a eseguire qualsiasi azione sulle risorse che appartengono all'account. Tuttavia, ciò non è valido per gli utenti IAM. Un utente IAM potrebbe disporre dell'accesso per creare una risorsa, ma le autorizzazioni dell'utente, anche per tale risorsa, sono limitate a quanto è stato concesso esplicitamente. Ciò significa che la semplice creazione di una risorsa, ad esempio di un ruolo IAM, non garantisce automaticamente l'autorizzazione per la modifica o l'eliminazione di tale ruolo. Inoltre, l'autorizzazione può essere revocata in qualsiasi momento dal proprietario dell'account o da un altro utente che dispone dell'accesso per gestire le autorizzazioni.

Controllo dell'accesso ai principali in un account specifico

È possibile concedere direttamente agli utenti IAM dell'account l'accesso alle risorse. Se gli utenti di un altro account devono accedere alle risorse, è possibile creare un ruolo IAM. Un ruolo è un'entità che include autorizzazioni, ma non è associata a un utente specifico. Gli utenti di altri account possono assumere quel ruolo e accedere alle risorse in base alle autorizzazioni assegnate al ruolo. Per ulteriori informazioni, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà.

> **Note**
>
> Alcuni servizi supportano le policy basate sulle risorse, come descritto in Policy basate sulle identità e policy basate su risorse (ad esempio Amazon S3, Amazon SNS e Amazon SQS). Per tali servizi, un'alternativa all'utilizzo dei ruoli consiste nel collegare una policy alla risorsa (bucket, argomento o coda) che si desidera condividere. La politica basata sulle risorse può specificare l' AWS account che dispone delle autorizzazioni per accedere alla risorsa.

Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag

Utilizza le informazioni nella sezione seguente per controllare chi può accedere agli utenti e ai ruoli IAM e a quali risorse gli utenti e i ruoli possono accedere. Per informazioni più generali ed esempi di controllo dell'accesso ad altre AWS risorse, incluse altre risorse IAM, consultaTagging delle risorse IAM.

> **Note**
>
> Per dettagli sulla distinzione tra maiuscole e minuscole per le chiavi dei tag e i valori delle chiavi dei tag, consulta Case sensitivity.

I tag possono essere collegati alla risorsa IAM, trasferiti nella richiesta o collegati al principale che effettua la richiesta. Un utente o ruolo IAM può essere sia una risorsa che un principale. Ad esempio, puoi scrivere una policy che consente a un utente di elencare i gruppi per un utente. Questa operazione è consentita solo se l'utente che effettua la richiesta (principale) ha lo stesso tag `project=blue` dell'utente che sta tentando di visualizzare. In questo esempio, l'utente può visualizzare l'appartenenza al gruppo per qualsiasi utente, incluso se stesso, purché stia lavorando sullo stesso progetto.

Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy. Quando crei una policy IAM, puoi utilizzare i tag IAM e la chiave di condizione tag associata per controllare l'accesso a quanto segue:

- Risorsa: controlla l'accesso alle risorse di utente o ruolo in base ai relativi tag. A tale scopo, usa la chiave di condizione aws:ResourceTag/***key-name*** per specificare quale coppia chiave-valore di tag deve essere associata alla risorsa. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse AWS.
- Richiesta: controlla quali tag possono essere passati in una richiesta IAM. A tale scopo, usa la chiave di condizione aws:RequestTag/***key-name*** per specificare quali tag possono essere aggiunti, modificati o rimossi da un utente o un ruolo IAM. Questa chiave viene utilizzata allo stesso modo per le risorse IAM e altre AWS risorse. Per ulteriori informazioni, consulta Controllo dell'accesso durante le richieste AWS.
- Principale: controlla le operazioni consentite alla persona che effettua la richiesta (il principale) in base ai tag collegati all'utente o al ruolo IAM di tale persona. A tale scopo, usa la chiave di condizione aws:PrincipalTag/***key-name*** per specificare quali tag devono essere allegati all'utente o al ruolo IAM prima che la richiesta sia consentita.
- Qualsiasi parte del processo di autorizzazione: utilizza la chiave aws: TagKeys condition per controllare se chiavi di tag specifiche possono essere utilizzate in una richiesta o da un principale. In questo caso, il valore chiave non è importante. Questa chiave si comporta in modo simile per IAM e altri AWS servizi. Tuttavia, quando aggiungi tag a un utente in IAM, questo controlla anche se il principale può effettuare la richiesta a qualsiasi servizio. Per ulteriori informazioni, consulta Controllo dell'accesso in base alle chiavi di tag.

È possibile creare una policy IAM utilizzando l'editor visivo, tramite JSON o importando una policy gestita esistente. Per informazioni dettagliate, vedi Creazione di policy IAM.

> **Note**
>
> Puoi anche passare tag di sessione quando assumi un ruolo IAM o esegui la federazione di un utente. Questi tag sono validi solo per la durata della sessione.

Controllo dell'accesso per i principali IAM

È possibile controllare le operazioni che il principale è autorizzato a eseguire in base ai tag collegati all'identità di tale persona.

Questo esempio mostra come creare una policy basata sull'identità che consenta a qualsiasi utente in questo account di visualizzare l'appartenenza al gruppo per qualsiasi utente, incluso sé stesso, purché stia lavorando sullo stesso progetto. Questa operazione è consentita solo quando il tag della risorsa dell'utente e il tag del principale hanno lo stesso valore per la chiave del tag `project`. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:ListGroupsForUser",
            "Resource": "arn:aws:iam::111222333444:user/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/project":
 "${aws:PrincipalTag/project}"}
            }
        }]
}
```

Controllo dell'accesso in base alle chiavi di tag

Puoi utilizzare i tag nelle tue policy IAM per controllare se chiavi di tag specifiche possono essere utilizzate in una richiesta o da un principale.

Questo esempio mostra come creare una policy basata sull'identità che consenta di rimuovere solo il tag con la chiave `temporary` da parte degli utenti. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:UntagUser",
        "Resource": "*",
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys": ["temporary"]}}
    }]
}
```

Controllo dell'accesso alle AWS risorse tramite tag

Puoi utilizzare i tag per controllare l'accesso alle tue AWS risorse che supportano l'etichettatura, incluse le risorse IAM. Puoi aggiungere i tag a utenti e ruoli IAM per controllare a cosa possono accedere. Per ulteriori informazioni su come applicare tag a utenti e ruoli IAM, consulta Tagging delle risorse IAM. Inoltre, puoi controllare l'accesso alle seguenti risorse IAM: policy gestite dal cliente, provider di identità IAM, profili delle istanze, certificati server e dispositivi MFA virtuali. Per visualizzare un tutorial per la creazione e il test di una policy che consente ai ruoli IAM con tag principali di accedere alle risorse con tag corrispondenti, consulta Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag. Utilizza le informazioni contenute nella sezione seguente per controllare l'accesso ad altre AWS risorse, incluse le risorse IAM, senza etichettare gli utenti o i ruoli IAM.

Prima di utilizzare i tag per controllare l'accesso alle tue AWS risorse, devi capire come AWS concedere l'accesso. AWS è composto da raccolte di risorse. Una istanza Amazon EC2 è una risorsa. Un bucket Amazon S3 è una risorsa. Puoi utilizzare l' AWS API AWS CLI, the o the AWS Management Console per eseguire un'operazione, come la creazione di un bucket in Amazon S3. In questo caso, invii una richiesta per tale operazione. La richiesta specifica un'operazione, una risorsa, un'entità principale (utente o ruolo), un account principale e qualsiasi altra informazione necessaria per la richiesta. Tutte queste informazioni forniscono il contesto.

AWS verifica quindi che tu (l'entità principale) sia autenticato (effettuato l'accesso) e autorizzato (disponi del permesso) a eseguire l'azione specificata sulla risorsa specificata. Durante l'autorizzazione, AWS controlla tutte le politiche che si applicano al contesto della richiesta.

La maggior parte delle politiche viene archiviata AWS come documenti JSON e specifica le autorizzazioni per le entità principali. Per ulteriori informazioni sui tipi di policy e i relativi utilizzi, consulta Policy e autorizzazioni in IAM.

AWS autorizza la richiesta solo se ogni parte della richiesta è consentita dalle politiche. Per visualizzare un diagramma e per ulteriori informazioni sull'infrastruttura IAM, consulta Funzionamento di IAM. Per ulteriori informazioni su come IAM determina se una richiesta è consentita, consulta Logica di valutazione delle policy.

I tag possono complicare questo processo perché possono essere collegati alla risorsa o trasferiti nella richiesta verso servizi che supportano il tagging. Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy. Per sapere se un AWS servizio supporta il controllo dell'accesso tramite tag, consulta AWS servizi che funzionano con IAM e cerca i servizi con Sì nella colonna ABAC. Scegli il nome del servizio per visualizzarne la documentazione sul controllo degli accessi e delle autorizzazioni.

Puoi quindi creare una policy IAM che consenta o rifiuti l'accesso a una risorsa in base al tag di quella risorsa. In quella policy, puoi utilizzare chiavi di condizione tag per controllare gli accessi a quanto segue:

- Risorsa: controlla l'accesso alle risorse del AWS servizio in base ai tag presenti su tali risorse. A tale scopo, utilizza la chiave di condizione ResourceTag/***key-name*** per determinare se consentire l'accesso alla risorsa in base ai tag allegati alla risorsa.
- Risorsa: controlla quali tag possono essere passati in una richiesta. Per fare ciò, usa la chiave di condizione aws:RequestTag/***key-name*** per specificare quali coppie di tag chiave-valore possono essere passate in una richiesta di taggare una risorsa. AWS
- Qualsiasi parte del processo di autorizzazione: usa la chiave aws: TagKeys condition per controllare se in una richiesta possono essere presenti chiavi di tag specifiche.

È possibile creare una policy IAM visivamente, utilizzando JSON o importando una policy gestita esistente. Per informazioni dettagliate, vedi Creazione di policy IAM.

> **Note**
>
> Alcuni servizi consentono agli utenti di specificare tag durante la creazione della risorsa, se dispongono delle autorizzazioni per utilizzare l'operazione che crea la risorsa.

Controllo dell'accesso alle risorse AWS

Puoi utilizzare le condizioni nelle tue policy IAM per controllare l'accesso alle AWS risorse in base ai tag di quella risorsa. Puoi eseguire questa operazione utilizzando la chiave di condizione `aws:ResourceTag/tag-key` globale o una chiave specifica del servizio. Alcuni servizi, supportano solo la versione specifica del servizio di questa chiave e non la versione globale.

> **Warning**
>
> Non cercare di controllare chi può passare un ruolo assegnando tag al ruolo e utilizzando la chiave di condizione `ResourceTag` in una policy con l'operazione `iam:PassRole`. Questo approccio non produce risultati affidabili. Per ulteriori informazioni sulle autorizzazioni richieste per trasferire un ruolo a un servizio, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS.

Questo esempio mostra come creare una policy basata sull'identità che consenta di avviare o arrestare le istanze Amazon EC2. Queste operazioni sono consentite solo se il tag dell'istanza `Owner` ha il valore del nome utente. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
```

```
}
```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard-roe` prova ad avviare un'istanza Amazon EC2, l'istanza deve avere il tag `0wner=richard-roe` o `owner=richard-roe`. In caso contrario, gli verrà negato l'accesso. La chiave di tag `0wner` corrisponde sia a `0wner` sia a `owner` perché i nomi delle chiavi di condizione non distinguono tra maiuscole e minuscole. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Condition.

Questo esempio mostra come creare una policy basata sull'identità che utilizza il tag del principale `team` nell'ARN della risorsa. La policy concede l'autorizzazione per eliminare le code del servizio di coda semplice Amazon (Amazon SQS), ma solo se il nome della coda inizia con il nome del team seguito da `-queue`. Ad esempio, `qa-queue` se `qa` è il nome del team per il tag del principale `team`.

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Sid": "AllQueueActions",
        "Effect": "Allow",
        "Action": "sqs:DeleteQueue",
        "Resource": "arn:aws:sqs:us-east-2::${aws:PrincipalTag/team}-queue"
      }
}
```

Controllo dell'accesso durante le richieste AWS

Puoi utilizzare le condizioni nelle tue policy IAM per controllare quali coppie chiave-valore di tag possono essere passate in una richiesta che applica tag a una AWS risorsa.

Questo esempio mostra come creare una policy basata sull'identità che consenta di utilizzare l'operazione `CreateTags` di Amazon EC2 per assegnare tag a un'istanza. Puoi collegare i tag solo se il tag contiene la chiave `environment` e i valori `production` o `preprod`. Se lo desideri, puoi utilizzare il modificatore `ForAllValues` con la chiave di condizione `aws:TagKeys` per indicare che nella richiesta è ammessa solo la chiave `environment`. In questo modo gli utenti smetteranno di includere altre chiavi ad esempio utilizzando per errore `Environment` invece di `environment`.

```
{
    "Version": "2012-10-17",
    "Statement": {
```

```
        "Effect": "Allow",
        "Action": "ec2:CreateTags",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/environment": [
                    "preprod",
                    "production"
                ]
            },
            "ForAllValues:StringEquals": {"aws:TagKeys": "environment"}
        }
    }
}
```

Controllo dell'accesso in base alle chiavi di tag

Puoi utilizzare una condizione nelle policy IAM per controllare se determinate chiavi di tag possono essere utilizzate in una richiesta.

Quando utilizzi le policy per controllare l'accesso tramite tag, ti consigliamo di utilizzare la chiave `aws:TagKeys`condition. AWS i servizi che supportano i tag potrebbero consentire di creare più nomi di chiavi di tag che differiscono solo in base alle maiuscole e minuscole, ad esempio etichettare un'istanza `stack=production` Amazon EC2 con e. `Stack=test` Nelle condizioni delle policy i nomi delle chiavi non distinguono tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe. Per impedire la duplicazione dei tag con una chiave che cambia solo per le dimensioni dei caratteri, utilizza la condizione `aws:TagKeys` per definire le chiavi di tag applicabili dagli utenti, o usa le policy di tag disponibili con AWS Organizations. Per ulteriori informazioni, consulta Policy delle tag nella Guida per l'utente di Organizations.

Questo esempio mostra come creare una policy basata sull'identità che consenta di creare e assegnare tag a un secreto di Secrets Manager, ma solo con le chiavi di tag `environment` o `cost-center`. La condizione `Null` garantisce che la condizione sia `false` in assenza di tag nella richiesta.

```
{
        "Effect": "Allow",
        "Action": [
```

```
            "secretsmanager:CreateSecret",
            "secretsmanager:TagResource"
        ],
        "Resource": "*",
        "Condition": {
            "Null": {
                "aws:TagKeys": "false"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "environment",
                    "cost-center"
                ]
            }
        }
}
```

Accesso alle risorse multi-account in IAM

Per alcuni AWS servizi, puoi concedere l'accesso a più account alle tue risorse utilizzando IAM. A tale scopo, è possibile collegare una policy direttamente alla risorsa da condividere oppure utilizzare un ruolo come proxy.

Per condividere direttamente la risorsa, la risorsa da condividere deve supportare le policy basate su risorse. A differenza di una policy basata su identità per un ruolo, una policy basata su risorse specifica chi (quale principale) può accedere a tale risorsa.

Utilizza un ruolo come proxy quando desideri accedere a risorse di un altro account che non supportano le policy basate su risorse.

Per ulteriori dettagli sulle differenze tra questi tipi di policy, consulta la sezione Policy basate sulle identità e policy basate su risorse.

> **Note**
>
> I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, poniamo che tu abbia un account nella Regione Stati Uniti occidentali (California settentrionale) nella partizione `aws` standard. Hai anche un account in Cina nella partizione `aws-cn`. Non puoi utilizzare una politica basata sulle risorse nel tuo account in Cina per consentire l'accesso agli utenti del tuo account standard. AWS

Accesso multi-account tramite ruoli

Non tutti i AWS servizi supportano politiche basate sulle risorse. Per questi servizi, puoi utilizzare i ruoli IAM multi-account per centralizzare la gestione delle autorizzazioni quando fornisci l'accesso multi-account a più servizi. Un ruolo IAM su più account è un ruolo IAM che include una policy di fiducia che consente ai responsabili IAM di un altro AWS account di assumere il ruolo. In poche parole, puoi creare un ruolo in un AWS account che delega autorizzazioni specifiche a un altro account. AWS

Per informazioni sul collegamento di una policy a un'identità IAM, consulta Gestione di policy IAM.

> **Note**
>
> Quando un principale passa a un ruolo per utilizzare temporaneamente le rispettive autorizzazioni, rinuncia alle autorizzazioni originali e assume quelle assegnate al ruolo che ha assunto.

Diamo un'occhiata al processo complessivo prendendo in esame un software di un partner APN che deve accedere a un account cliente.

1. Il cliente crea un ruolo IAM nel proprio account con una policy IAM che consente l'accesso alle risorse Amazon S3 richieste dal partner APN. In questo esempio, il nome del ruolo è `APNPartner`.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "s3:*",
               "Resource": [
                   "arn:aws:s3:::bucket-name"
               ]
           }
       ]
   }
   ```

2. Quindi, il cliente specifica che il ruolo può essere assunto dall' AWS account del partner fornendo l' Account AWS ID del partner APN nella politica di fiducia relativa al ruolo. `APNPartner`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::APN-account-ID:role/APN-user-name"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

3. Il cliente fornisce il nome della risorsa Amazon (ARN) del ruolo al partner APN. L'ARN è il nome completo del ruolo.

```
arn:aws:iam::APN-ACCOUNT-ID:role/APNPartner
```

> **Note**
>
> Ti consigliamo di utilizzare un ID esterno in situazioni multi-tenant. Per informazioni dettagliate, vedi Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.

4. Quando il software del partner APN deve accedere all'account del cliente, chiama l'AssumeRoleAPI AWS Security Token Service con l'ARN del ruolo nell'account del cliente. STS restituisce una AWS credenziale temporanea che consente al software di svolgere il proprio lavoro.

Per un altro esempio di concessione dell'accesso multi-account utilizzando i ruoli, consulta la sezione Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà. Puoi anche seguire il Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM.

Accesso multi-account utilizzando policy basate su risorse

Quando un account accede a una risorsa tramite un altro account utilizzando una policy basata su risorse, il principale continua a utilizzare l'account attendibile e non deve rinunciare alle proprie autorizzazioni per ricevere quelle del ruolo. In altre parole, il principale ha accesso alle risorse

dell'account attendibile e contemporaneamente alla risorsa nell'account che concede fiducia. Questa funzione è utile per attività come la copia di informazioni da o verso la risorsa condivisa nell'altro account.

I principi che è possibile specificare in una policy basata sulle risorse includono account, utenti IAM, utenti federati, ruoli IAM, sessioni con ruolo presunto o servizi. AWS Per ulteriori informazioni, consulta la sezione Specifica di un principale.

Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta la sezione Identificazione di risorse condivise con un'entità esterna.

L'elenco seguente include alcuni dei AWS servizi che supportano le politiche basate sulle risorse. Per un elenco completo del numero crescente di AWS servizi che supportano l'associazione di politiche di autorizzazione alle risorse anziché ai principali, consulta AWS servizi che funzionano con IAM e cerca i servizi con Sì nella colonna Resource Based.

- Bucket Amazon S3: la policy è collegata al bucket, ma controlla l'accesso sia al bucket sia agli oggetti in esso contenuti. Per ulteriori informazioni, consulta Controllo accessi nella Guida per l'utente di Amazon Simple Storage Service. In alcuni casi, può essere opportuno utilizzare ruoli per l'accesso per più account ad Amazon S3. Per ulteriori informazioni, consulta le spiegazioni passo per passo di esempio nella Guida per l'utente di Amazon Simple Storage Service.
- Argomenti Amazon Simple Notification Service (Amazon SNS): per ulteriori informazioni, consulta la sezione Casi di esempio per il controllo degli accessi Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.
- Code di Amazon Simple Queue Service (Amazon SQS): per ulteriori informazioni, consulta Appendice: La sintassi della policy di accesso nellaGuida per gli sviluppatori di Amazon Simple Queue Service.

Delega delle AWS autorizzazioni in una politica basata sulle risorse

Se una risorsa concede le autorizzazioni ai principali nell'account, puoi delegare tali autorizzazioni a identità IAM specifiche. Le identità sono utenti, gruppi di utenti o ruoli nell'account. Per delegare le autorizzazioni, collegare una policy all'identità. È possibile concedere fino al numero massimo di autorizzazioni consentite dall'account proprietario della risorsa.

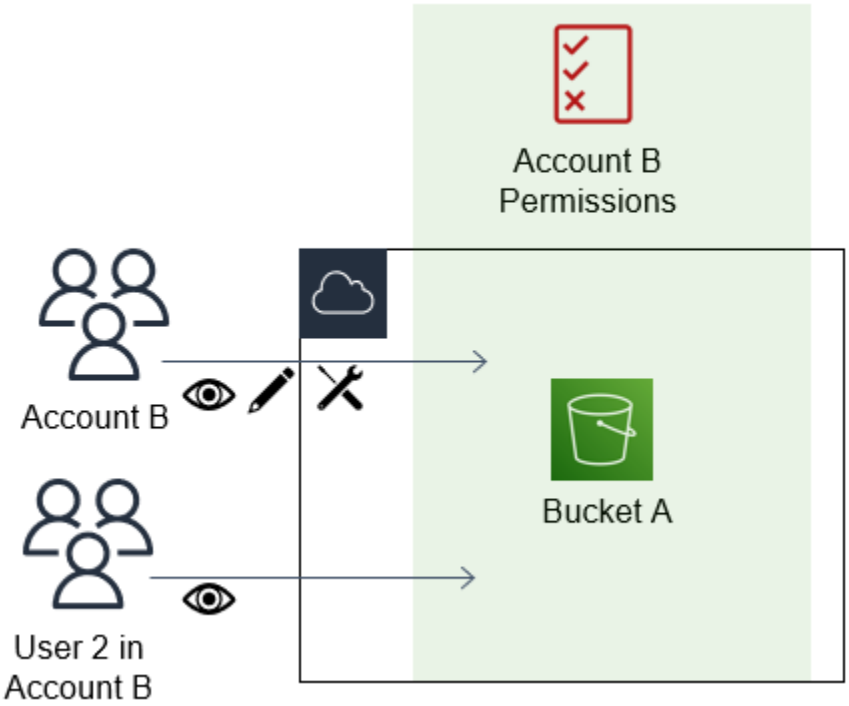
> **Important**
>
> Nell'accesso multi-account, il principale deve disporre della condizione `Allow` nella policy di identità e nella policy basata su risorse.

Si supponga che una policy basata sulle risorse consenta a tutti i principali nell'account l'accesso amministrativo completo a una risorsa. Quindi puoi delegare l'accesso completo, l'accesso in sola lettura o qualsiasi altro accesso parziale ai principali del tuo account. AWS In alternativa, se la policy basata sulle risorse consente solo le autorizzazioni per la presentazione di elenchi, è possibile delegare solo l'accesso all'elenco. Se si tenta di delegare più autorizzazioni rispetto a quelle possedute dall'account, i principali avranno comunque solo accesso all'elenco.

Per ulteriori informazioni su come vengono prese queste decisioni, consulta la sezione Determinare se una richiesta è consentita o rifiutata all'interno di un account.

> **Note**
>
> I ruoli IAM e le policy basate sulle risorse delegano l'accesso tra account solo all'interno di una singola partizione. Ad esempio, non è possibile aggiungere l'accesso tra account tra un account nella partizione `aws` standard e un account nella partizione `aws-cn`.

Ad esempio, si supponga di gestire `AccountA` e `AccountB`. Nell'AccountA, disponi di un bucket Amazon S3 denominato `BucketA`.



1. Colleghi una policy basata su risorse ad `BucketA` che consente a tutti i principali nell'AccountB l'accesso completo agli oggetti nel bucket. Possono creare, leggere o eliminare qualsiasi oggetto in tale bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PrincipalAccess",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::AccountB:root"},
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::BucketA/*"
            }
        ]
}
```

   AccountA fornisce all'AccountB l'accesso completo al BucketA denominando AccountB come un principale nella policy basata su risorse. Di conseguenza, l'AccountB è autorizzato a eseguire qualsiasi operazione nel BucketA e l'amministratore dell'AccountB può delegare l'accesso ai propri utenti nell'AccountB.

L'utente root dell'AccountB dispone di tutte le autorizzazioni concesse all'account. Pertanto, l'utente root dispone di accesso completo al BucketA.

2. Nell'AccountB, collega una policy all'utente IAM denominato User2. Tale policy consente all'utente l'accesso in sola lettura agli oggetti nel BucketA. Ciò significa che User2 può visualizzare gli oggetti, ma non crearli, modificarli o eliminarli.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect" : "Allow",
            "Action" : [
                "s3:Get*",
                "s3:List*" ],
                "Resource" : "arn:aws:s3:::BucketA/*"
        }
    ]
}
```

Il livello massimo di accesso che AccountB è in grado di delegare è il livello di accesso concesso all'account. In questo caso, la policy basata su risorse ha concesso l'accesso completo all'AccountB, ma User2 dispone solo dell'accesso di sola lettura.

L'amministratore dell'AccountB non concede l'accesso a User1. Per impostazione predefinita, gli utenti non dispongono di autorizzazioni ad eccezione di quelle concesse in modo esplicito, pertanto User1 non ha accesso al BucketA.

IAM valuta le autorizzazioni di un principale nel momento in cui il principale effettua una richiesta. Se usi i caratteri jolly (*) per consentire agli utenti l'accesso completo alle tue risorse, i mandanti possono accedere a tutte le risorse a cui ha accesso il tuo account. AWS Ciò vale anche per le risorse che vengono aggiunte o a cui si ha accesso dopo aver creato le policy dell'utente.

Nell'esempio precedente, se l'AccountB avesse collegato a User2 una policy che concedeva l'accesso completo a tutte le risorse in tutti gli account, User2 avrebbe automaticamente avuto accesso a qualsiasi risorsa alla quale ha accesso l'AccountB. Ciò include l'accesso al BucketA e l'accesso a qualsiasi altra risorsa concesso dalle policy basate su risorse nell'AccountA.

Per ulteriori informazioni sugli usi complessi dei ruoli, come la concessione dell'accesso ad applicazioni e servizi, consulta la sezione Scenari comuni per ruoli: utenti, applicazioni e servizi.

> **Important**
>
> Concedere l'accesso solo a entità attendibili e fornire il livello minimo di accesso necessario. Ogni volta che l'entità fidata è un altro AWS account, a qualsiasi responsabile IAM può essere concesso l'accesso alla tua risorsa. L' AWS account fidato può delegare l'accesso solo nella misura in cui gli è stato concesso l'accesso; non può delegare un accesso maggiore di quello concesso all'account stesso.

Per ulteriori informazioni sulle autorizzazioni, le policy e il linguaggio di policy di autorizzazioni che è possibile utilizzare per scrivere policy, consultare Gestione degli accessi AWS alle risorse.

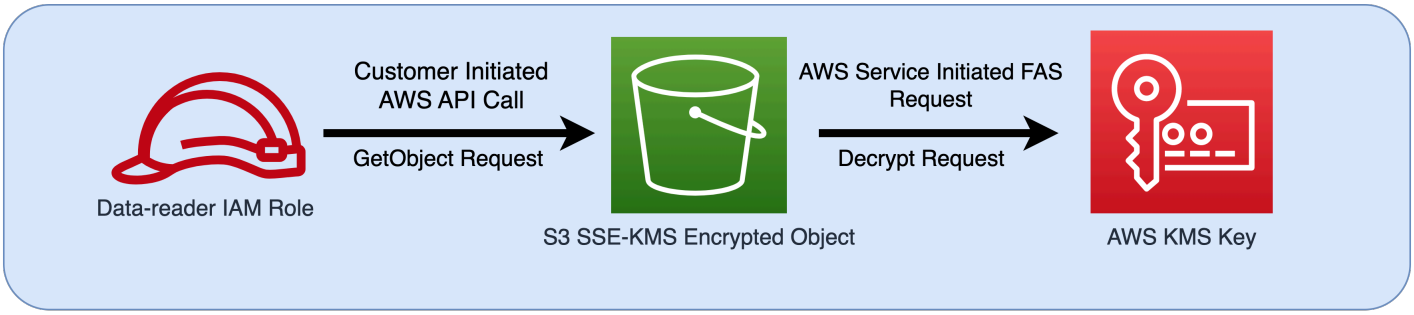
Inoltro delle sessioni di accesso

Le sessioni di accesso inoltrato (FAS) sono una tecnologia IAM utilizzata dai AWS servizi per trasmettere identità, autorizzazioni e attributi di sessione quando un AWS servizio effettua una richiesta per conto dell'utente. FAS utilizza le autorizzazioni dell'identità che chiama un AWS servizio, combinate con l'identità di un AWS servizio per effettuare richieste ai servizi downstream. Le richieste FAS vengono inviate ai AWS servizi per conto di un responsabile IAM solo dopo che un servizio ha ricevuto una richiesta che richiede interazioni con altri AWS servizi o risorse per essere completata. Quando viene effettuata una richiesta FAS:

- Il servizio che riceve la richiesta iniziale da un principale IAM controlla le autorizzazioni di tale principale IAM.
- Anche il servizio che riceve la conseguente richiesta FAS controlla le autorizzazioni dello stesso principale IAM.

Ad esempio, FAS viene utilizzato da Amazon S3 per effettuare chiamate AWS Key Management Service per decrittografare un oggetto quando SSE-KMS è stato utilizzato per crittografarlo. Quando si scarica un oggetto crittografato SSE-KMS, un ruolo denominato data-reader chiama l'oggetto GetObject su Amazon S3 e non chiama direttamente. AWS KMS Dopo aver ricevuto la GetObject richiesta e autorizzato il lettore di dati, Amazon S3 effettua quindi una richiesta FAS AWS KMS a per decrittografare l'oggetto Amazon S3. Quando KMS riceve la richiesta FAS, controlla le autorizzazioni del ruolo e autorizza la richiesta di decrittografia solamente se data-reader

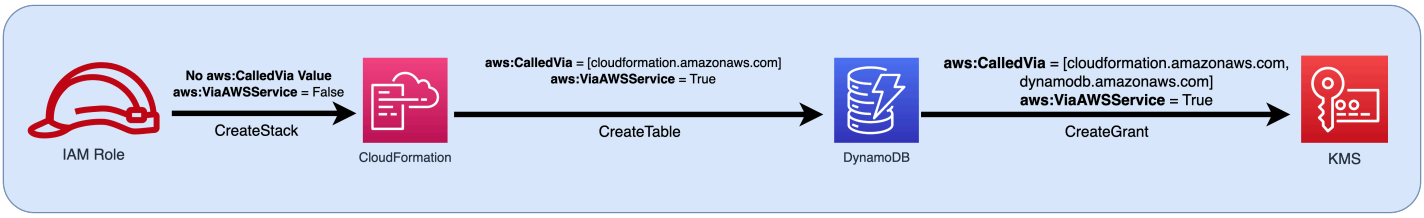
dispone delle autorizzazioni corrette sulla chiave KMS. Le richieste ad Amazon S3 AWS KMS sono autorizzate utilizzando le autorizzazioni del ruolo e hanno esito positivo solo se data-reader dispone delle autorizzazioni sia per l'oggetto Amazon S3 che per la chiave KMS. AWS



> **Note**
>
> I servizi che hanno ricevuto una richiesta FAS possono a loro volta effettuare richieste FAS aggiuntive. In questi casi, il principale che esegue la richiesta deve disporre delle autorizzazioni per tutti i servizi chiamati da FAS.

Richieste FAS e condizioni delle policy IAM

Quando vengono effettuate richieste FAS, le chiavi di condizione Leggi: CalledVia, aws: CalledVia Primo e aws: CalledVia Ultimo vengono compilate con il principale di servizio del servizio che ha avviato la chiamata FAS. Il valore della chiave di condizione AWS: via AWSService viene impostato su `true` ogni volta che viene effettuata una richiesta FAS. Nel diagramma seguente, per la richiesta a direct non è impostata CloudFormation alcuna chiave o condizionale. `aws:CalledVia` `aws:ViaAWSService` Quando CloudFormation e DynamoDB effettuano richieste FAS downstream per conto del ruolo, i valori per queste chiavi di condizione vengono compilati.



Per consentire l'esecuzione di una richiesta FAS che altrimenti verrebbe negata da un'istruzione di una policy di rifiuto con una chiave di condizione che verifica gli indirizzi IP o i VPC di origine, è necessario utilizzare le chiavi di condizione per impostare un'eccezione per le richieste FAS nella policy di rifiuto. Questa operazione può essere eseguita per tutte le richieste FAS utilizzando la

chiave di condizione `aws:ViaAWSService`. Per consentire solo a AWS servizi specifici di effettuare richieste FAS, usa. `aws:CalledVia`

> **Important**
>
> Quando viene effettuata una richiesta FAS in seguito a una richiesta iniziale effettuata tramite un endpoint VPC, i valori delle chiavi di condizione per `leggi: SourceVpce`, `come: SourceVpc` e `aws: VpcSource Ip` della richiesta iniziale non vengono utilizzati nelle richieste FAS. Quando si scrivono policy utilizzando `aws:VPCSourceIP` o `aws:SourceVPCE` per concedere l'accesso in modo condizionale, è inoltre necessario utilizzare `aws:ViaAWSService` o `aws:CalledVia` per consentire le richieste FAS. Quando viene effettuata una richiesta FAS dopo che una richiesta iniziale è stata ricevuta da un endpoint di AWS servizio pubblico, le richieste FAS successive verranno effettuate con lo stesso `aws:SourceIP` valore della chiave di condizione.

Esempio: consentire ad Amazon S3 l'accesso da un VPC o tramite FAS

Nel seguente esempio di policy IAM, le richieste Amazon S3 e GetObject Athena sono consentite solo se provengono da endpoint VPC collegati a *`example_vpc`* o se la richiesta è una richiesta FAS effettuata da Athena.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "OnlyAllowMyIPs",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject*",
        "athena:StartQueryExecution",
        "athena:GetQueryResults",
        "athena:GetWorkGroup",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceVPC": [
```

```
            "example_vpc"
          ]
        }
      }
    },
    {
      "Sid": "OnlyAllowFAS",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject*"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": "athena.amazonaws.com"
        }
      }
    }
  ]
}
```

Per ulteriori esempi di utilizzo delle chiavi di condizione per consentire l'accesso FAS, consulta il repository di esempi di policy per implementare un perimetro di dati.

Esempi di policy basate su identità IAM

Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un responsabile IAM (utente o ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata AWS come documenti JSON allegati a un'identità IAM (utente, gruppo di utenti o ruolo). Le policy basate sulle identità includono policy gestite da AWS , policy gestite dal cliente e policy inline. Per ulteriori informazioni su come creare una policy IAM utilizzando questi documenti di policy JSON, consulta the section called “Creazione di policy utilizzando l'editor JSON”.

Per impostazione predefinita, tutte le richieste vengono rifiutate, pertanto è necessario fornire l'accesso a servizi, azioni e risorse da rendere disponibili per l'identità. Se desideri consentire l'accesso anche per completare le operazioni specificate nella console IAM, dovrai fornire ulteriori autorizzazioni.

La seguente libreria di policy può aiutarti a definire le autorizzazioni per le tue identità IAM. Una volta trovata la policy desiderata, seleziona view this policy (visualizza la policy) per consultare il JSON della policy. Puoi utilizzare il documento di policy JSON come modello per le tue policy.

> **Note**
>
> Per inviare una policy e includerla in questa guida di riferimento, utilizza il pulsante Feedback in fondo a questa pagina.

Politiche di esempio: AWS

- Consente l'accesso in un intervallo di date specifico. (Visualizzare questa policy).
- Consente di abilitare e disabilitare AWS le regioni. (Visualizzare questa policy).
- Consente agli utenti autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente un accesso specifico quando utilizzi MFA durante un determinato intervallo di date. (Visualizzare questa policy).
- Consente agli utenti di gestire le proprie credenziali nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire la propria password nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Consente agli utenti di gestire la propria password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza. (Visualizzare questa policy).
- Nega l'accesso a in AWS base alla regione richiesta. (Visualizzare questa policy).
- Nega l'accesso a in AWS base all'indirizzo IP di origine. (Visualizzare questa policy).

Politica di esempio: AWS Data Exchange

- Nega l'accesso alle risorse Amazon S3 al di fuori del tuo account, tranne AWS Data Exchange. (Visualizzare questa policy).

Politiche di esempio: AWS Data Pipeline

- Rifiuta l'accesso alle pipeline non create dall'utente (Visualizzare questa policy).

Policy di esempio: Amazon DynamoDB

- Consente l'accesso a una specifica tabella Amazon DynamoDB (Visualizza questa policy).
- Consente l'accesso ad attributi Amazon DynamoDB specifici (Visualizza questa policy).
- Consente l'accesso a livello di elemento ad Amazon DynamoDB in base a un ID Amazon Cognito (Visualizza questa policy).

Policy di esempio: Amazon EC2

- Consente il collegamento o il distacco di volumi Amazon EBS a istanze Amazon EC2 in base ai tag (Visualizza questa policy).
- Consente l'avvio di istanze Amazon EC2 in una sottorete specifica, in modo programmatico e nella console (Visualizza questa policy).
- Consente la gestione dei gruppi di sicurezza Amazon EC2 associati a un VPC specifico, a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'avvio e l'arresto di istanze Amazon EC2 taggate dall'utente, a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'avvio o l'arresto di istanze Amazon EC2 basate sui tag del principale e della risorsa, a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'avvio o l'arresto di istanze Amazon EC2 quando i tag del principale e della risorsa corrispondono (Visualizza questa policy).
- Consente l'accesso completo ad Amazon EC2 entro una regione specifica, a livello di programmazione e nella console. (Visualizzare questa policy).
- Consente l'avvio o l'arresto di un'istanza Amazon EC2 specifica e la modifica di un gruppo di sicurezza specifico, in modo programmatico e nella console (Visualizza questa policy).
- Rifiuta l'accesso a operazioni Amazon EC2 specifiche senza MFA (Visualizza questa policy).
- Limita la terminazione di istanze Amazon EC2 a un intervallo specifico di intervalli IP (Visualizza questa policy).

Politiche di esempio: AWS Identity and Access Management (IAM)

- Consente l'accesso all'API del simulatore di policy (Visualizzare questa policy).
- Consente l'accesso alla console del simulatore di policy (Visualizzare questa policy).
- Consente l'assunzione di qualsiasi ruolo che dispone di un tag specifico, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente e nega l'accesso a più servizi, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'aggiunta di un tag specifico a un utente IAM con un altro tag specifico, a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'aggiunta di un tag specifico a qualsiasi utente o ruolo IAM, a livello di programmazione e nella console (Visualizza questa policy).
- Consente la creazione di un nuovo utente solo con tag specifici (Visualizzare questa policy).
- Consente la generazione e il recupero di report delle credenziali IAM (Visualizzare questa policy).
- Consente la gestione dell'appartenenza a un gruppo, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente la gestione di un tag specifico (Visualizzare questa policy).
- Consente di passare un ruolo IAM a un servizio specifico (Visualizza questa policy).
- Consente l'accesso in sola lettura alla console IAM senza la creazione di report (Visualizza questa policy).
- Consente l'accesso in sola lettura alla console IAM (Visualizza questa policy).
- Consente a utenti specifici di gestire un gruppo, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'impostazione di requisiti della password dell'account, a livello di programmazione e nella console (Visualizzare questa policy).
- Consente l'utilizzo dell'API del simulatore di policy per gli utenti con un percorso specifico (Visualizzare questa policy).
- Consente l'utilizzo della console del simulatore di policy per gli utenti con un percorso specifico (Visualizzare questa policy).
- Consente agli utenti IAM di gestire in modo autonomo un dispositivo MFA (Visualizzare questa policy).
- Consente agli utenti IAM di impostare le credenziali a livello di programmazione e nella console. (Visualizzare questa policy).

- Consente di visualizzare le informazioni sull'ultimo accesso al servizio per una AWS Organizations policy nella console IAM. (Visualizzare questa policy).
- Limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM (Visualizza questa policy).
- Consente l'accesso alle policy IAM solo nel tuo account. Visualizza questa policy.

Politiche di esempio: AWS Lambda

- Consente a una AWS Lambda funzione di accedere a una tabella Amazon DynamoDB (Visualizza questa policy).

Policy di esempio: Amazon RDS

- Consente l'accesso completo al database Amazon RDS in una regione specifica. (Visualizzare questa policy).
- Consente il ripristino dei database Amazon RDS, in modo programmatico e nella console (Visualizza questa policy)
- Consente ai proprietari di tag l'accesso completo alle risorse Amazon RDS che hanno taggato (Visualizza questa policy).

Policy di esempio: Amazon S3

- Consente a un utente Amazon Cognito di accedere a oggetti del proprio bucket Amazon S3 (Visualizza questa policy)
- Consente agli utenti federati di accedere alla loro directory home in Amazon S3 a livello di programmazione e nella console (Visualizza questa policy).
- Consente l'accesso S3 completo, ma nega esplicitamente l'accesso al bucket di produzione se l'amministratore non ha effettuato l'accesso utilizzando MFA negli ultimi trenta minuti (Visualizzare questa policy).
- Consente agli utenti IAM di accedere alla propria directory home in Amazon S3, in modo programmatico e nella console (Visualizza questa policy).
- Consente a un utente di gestire un singolo bucket Amazon S3 e nega ogni altra AWS azione e risorsa (Visualizza questa policy).

- Consente un accesso di tipo `Read` e `Write` a un bucket Amazon S3 specifico (Visualizza questa policy).
- Consente un accesso di tipo `Read` e `Write` a un bucket Amazon S3 specifico, in modo programmatico e nella console (Visualizza questa policy).

AWS: consente l'accesso in base alla data e all'ora

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso alle operazioni in base alla data e all'ora. Questa policy limita l'accesso alle operazioni che si verificano tra il 1° aprile 2020 e il 30 giugno 2020 (UTC), inclusi. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni sull'utilizzo di condizioni multiple all'interno di un blocco `Condition` di una policy IAM, consultare Valori multipli in una condizione

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "service-prefix:action-name",
            "Resource": "*",
            "Condition": {
                "DateGreaterThan": {"aws:CurrentTime": "2020-04-01T00:00:00Z"},
                "DateLessThan": {"aws:CurrentTime": "2020-06-30T23:59:59Z"}
            }
        }
    ]
}
```

> **Note**
>
> Non è possibile utilizzare una variabile di policy con l'operatore di condizione Date. Per ulteriori informazioni, consulta la sezione Elemento condizione

AWS: consente di abilitare e disabilitare le regioni AWS

Questo esempio mostra come creare una policy basata sull'identità che consenta a un amministratore di abilitare e disabilitare la regione Asia Pacifico (Hong Kong) (ap-east-1). Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Questa impostazione viene visualizzata nella pagina Account settings (Impostazioni dell'account) nella AWS Management Console. Questa pagina include informazioni sensibili a livello di account, che devono essere visualizzate e gestite solo da amministratori dell'account. Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Important**
>
> Non è possibile abilitare o disabilitare le regioni abilitate per impostazione predefinita. Puoi includere solo le regioni disabilitate per impostazione predefinita. Per ulteriori informazioni, consulta Gestione delle regioni AWS nella Riferimenti generali di AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableDisableHongKong",
            "Effect": "Allow",
            "Action": [
                "account:EnableRegion",
                "account:DisableRegion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"account:TargetRegion": "ap-east-1"}
            }
        },
        {
            "Sid": "ViewConsole",
            "Effect": "Allow",
            "Action": [
                "account:ListRegions"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM autenticati tramite l'autenticazione a più fattori (MFA) di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Questa pagina della AWS Management Console mostra informazioni sull'account, come l'ID account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i dispositivi MFA, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare tutte le informazioni sulla pagina. Richiede inoltre che l'utente si configuri e si autentichi tramite MFA prima di eseguire qualsiasi altra operazione in. AWS Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per informazioni su come gli utenti possono accedere alla pagina delle credenziali di sicurezza, consulta. Come gli utenti IAM possono cambiare le proprie password (console)

> **Note**
>
> - Questo criterio di esempio non consente agli utenti di reimpostare una password durante AWS Management Console il primo accesso. Ti consigliamo di non concedere autorizzazioni ai nuovi utenti fino a quando non hanno effettuato l'accesso. Per ulteriori informazioni, consulta Come posso creare utenti IAM in modo sicuro?. Inoltre, ciò impedisce agli utenti con una password scaduta di reimpostare la password durante l'accesso. Per consentire questa operazione, aggiungere `iam:ChangePassword` e `iam:GetAccountPasswordPolicy` all'istruzione `DenyAllExceptListedIfNoMFA`. Tuttavia, non ti consigliamo di farlo perché consentire agli utenti di cambiare la password senza MFA può costituire un rischio per la sicurezza.
> - Se intendi utilizzare questa policy per l'accesso programmatico, devi chiamare `GetSessionToken` per l'autenticazione con l'MFA. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto da MFA.

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:
  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `ListVirtualMFADevices`: consente di visualizzare i dettagli di un dispositivo MFA virtuale abilitato per l'utente.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.
- L'istruzione `AllowManageOwnSigningCertificates` consente all'utente di caricare, aggiornare ed eliminare i propri certificati di firma.
- L'`AllowManageOwnSSHPublicKeys`istruzione consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per CodeCommit.
- L'`AllowManageOwnGitCredentials`istruzione consente all'utente di creare, aggiornare ed eliminare le proprie credenziali Git per CodeCommit.
- L'istruzione `AllowManageOwnVirtualMFADevice` consente all'utente di creare il proprio dispositivo virtuale MFA. L'ARN della risorsa in questa istruzione consente all'utente di creare un dispositivo MFA con qualsiasi nome, ma le altre istruzioni nella policy consentono all'utente soltanto di collegare il dispositivo all'utente correntemente registrato.
- L'istruzione `AllowManageOwnUserMFA` consente all'utente di visualizzare o gestire il dispositivo MFA virtuale, U2F o hardware per il proprio utente. L'ARN della risorsa in questa istruzione consente di accedere solo all'utente IAM dell'utente stesso. Gli utenti non possono visualizzare o gestire il dispositivo MFA per altri utenti.
- L'`DenyAllExceptListedIfNoMFA`istruzione nega l'accesso a tutte le azioni in tutti i AWS servizi, ad eccezione di alcune azioni elencate, ma solo se l'utente non ha effettuato l'accesso con MFA. L'istruzione utilizza una combinazione di `"Deny"` e `"NotAction"` per negare esplicitamente l'accesso a tutte le operazioni che non sono nell'elenco. Gli elementi elencati non sono negati o consentiti da questa istruzione. Tuttavia, le azioni sono consentite da altre istruzioni della policy.

Per ulteriori informazioni sulla logica di questa istruzione, vedere NotAction with Deny. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `Condition` dà esito negativo e questa istruzione non produce effetti. In questo caso, altre policy o dichiarazioni per l'utente determinano le autorizzazioni dell'utente.

Questa istruzione garantisce che quando l'utente non ha effettuato l'accesso con MFA può eseguire solo le operazioni elencate. Inoltre, possono eseguire le operazioni elencate, solo se un'altra istruzione o policy consente l'accesso a tali operazioni. Questo non consente a un utente di creare una password all'accesso, perché l'operazione `iam:ChangePassword` non deve essere consentita senza l'autorizzazione MFA.

La versione `...IfExists` dell'operatore `Bool` garantisce che se la chiave `leggi: MultiFactor AuthPresent` manca, la condizione restituisce true. Questo significa che a un utente che accede a un'API con le credenziali di lungo termine, ad esempio con una chiave di accesso, viene negato l'accesso alle operazioni API non IAM.

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo` e all'istruzione `DenyAllExceptListedIfNoMFA`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`. Inoltre, per consente a un utente di cambiare la password dalla propria pagina utente senza l'accesso tramite MFA, aggiungere l'operazione `iam:UpdateLoginProfile` all'istruzione `DenyAllExceptListedIfNoMFA`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowManageOwnPasswords",
            "Effect": "Allow",
            "Action": [
                "iam:ChangePassword",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:GetAccessKeyLastUsed"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSigningCertificates",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSigningCertificate",
                "iam:ListSigningCertificates",
                "iam:UpdateSigningCertificate",
                "iam:UploadSigningCertificate"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSSHPublicKeys",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSSHPublicKey",
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
```

```
            "Sid": "AllowManageOwnGitCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "iam:DeleteServiceSpecificCredential",
                "iam:ListServiceSpecificCredentials",
                "iam:ResetServiceSpecificCredential",
                "iam:UpdateServiceSpecificCredential"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice",
                "sts:GetSessionToken"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

AWS: consente l'accesso specifico tramite MFA entro date specifiche

Questo esempio mostra come creare una policy basata sull'identità che utilizzi più condizioni che vengono valutate in base a un operatore AND logico. Consente l'accesso completo al servizio denominato SERVICE-NAME-1 e l'accesso alle operazioni ACTION-NAME-A e ACTION-NAME-B nel servizio denominato SERVICE-NAME-2. Queste operazioni sono consentite solo quando l'utente è autenticato tramite l'autenticazione a più fattori (MFA). L'accesso è limitato alle operazioni effettuate tra il 1 luglio 2017 e il 31 dicembre 2017 (UTC), inclusi. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni sull'utilizzo di condizioni multiple all'interno di un blocco Condition di una policy IAM, consultare Valori multipli in una condizione

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "service-prefix-1:*",
            "service-prefix-2:action-name-a",
            "service-prefix-2:action-name-b"
        ],
        "Resource": "*",
        "Condition": {
            "Bool": {"aws:MultiFactorAuthPresent": true},
            "DateGreaterThan": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
            "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
        }
    }
}
```

AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM di gestire tutte le proprie credenziali nella pagina Credenziali di sicurezza. Questa AWS Management Console pagina mostra informazioni sull'account come l'ID dell'account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare le informazioni sulla pagina eccetto il dispositivo MFA dell'utente. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per informazioni su come gli utenti possono accedere alla pagina delle credenziali di sicurezza, consulta. Come gli utenti IAM possono cambiare le proprie password (console)

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:
  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `GetAccountSummary`: visualizza l'ID account e l'ID utente canonico dell'account.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.
- L'istruzione `AllowManageOwnSigningCertificates` consente all'utente di caricare, aggiornare ed eliminare i propri certificati di firma.
- L'`AllowManageOwnSSHPublicKeys`istruzione consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per. CodeCommit

- L'`AllowManageOwnGitCredentials`istruzione consente all'utente di creare, aggiornare ed eliminare le proprie credenziali Git per CodeCommit.

Questa policy non consente agli utenti di visualizzare o gestire i propri dispositivi MFA. Inoltre, non sono in grado di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungere le operazioni `iam:CreateLoginProfile`, `iam:DeleteLoginProfile`, `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:GetAccountSummary"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswords",
            "Effect": "Allow",
            "Action": [
                "iam:ChangePassword",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:GetAccessKeyLastUsed"
```

```
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSigningCertificates",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSigningCertificate",
                "iam:ListSigningCertificates",
                "iam:UpdateSigningCertificate",
                "iam:UploadSigningCertificate"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSSHPublicKeys",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSSHPublicKey",
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnGitCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "iam:DeleteServiceSpecificCredential",
                "iam:ListServiceSpecificCredentials",
                "iam:ResetServiceSpecificCredential",
                "iam:UpdateServiceSpecificCredential"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS: consente agli utenti IAM autenticati tramite MFA di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM autenticati tramite l'autenticazione a più fattori (MFA) di gestire il proprio dispositivo MFA nella pagina Credenziali di sicurezza. Questa AWS Management Console pagina mostra le informazioni sull'account e sull'utente, ma l'utente può solo visualizzare e modificare il proprio dispositivo MFA. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

> **Note**
>
> Se un utente IAM con questa policy non è autenticato tramite MFA, questa policy nega l'accesso a tutte le AWS azioni tranne quelle necessarie per l'autenticazione tramite MFA. Per utilizzare l' AWS API AWS CLI and, gli utenti IAM devono prima recuperare il proprio token MFA utilizzando AWS STS GetSessionTokenl'operazione e quindi utilizzare quel token per autenticare l'operazione desiderata. Altre policy, ad esempio le policy basate sulle risorse o altre policy basate sull'identità, possono consentire operazioni in altri servizi. Questa policy negherà tale accesso se l'utente IAM non dispone dell'autenticazione MFA.

Per informazioni su come gli utenti possono accedere alla pagina delle credenziali di sicurezza, consulta. Come gli utenti IAM possono cambiare le proprie password (console)

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare i dettagli di un dispositivo MFA virtuale abilitato per l'utente. Questa autorizzazione deve essere nella propria dichiarazione perché non specifica un ARN della risorsa. È necessario invece specificare `"Resource" : "*"`.
- L'istruzione `AllowManageOwnVirtualMFADevice` consente all'utente di creare il proprio dispositivo virtuale MFA. L'ARN della risorsa in questa istruzione consente all'utente di creare un dispositivo MFA con qualsiasi nome, ma le altre istruzioni nella policy consentono all'utente soltanto di collegare il dispositivo all'utente correntemente registrato.
- L'istruzione `AllowManageOwnUserMFA` consente all'utente di visualizzare o gestire il proprio dispositivo MFA virtuale, U2F o hardware. L'ARN della risorsa in questa istruzione consente di accedere solo all'utente IAM dell'utente stesso. Gli utenti non possono visualizzare o gestire il dispositivo MFA per altri utenti.

- L'`DenyAllExceptListedIfNoMFA`istruzione nega l'accesso a tutte le azioni in tutti i AWS servizi, ad eccezione di alcune azioni elencate, ma solo se l'utente non ha effettuato l'accesso con MFA. L'istruzione utilizza una combinazione di `"Deny"` e `"NotAction"` per negare esplicitamente l'accesso a tutte le operazioni che non sono nell'elenco. Gli elementi elencati non sono negati o consentiti da questa istruzione. Tuttavia, le azioni sono consentite da altre istruzioni della policy. Per ulteriori informazioni sulla logica di questa istruzione, vedere NotAction with Deny. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `Condition` dà esito negativo e questa istruzione non produce effetti. In questo caso, altre policy o dichiarazioni per l'utente determinano le autorizzazioni dell'utente.

  Questa istruzione garantisce che quando l'utente non ha effettuato l'accesso con MFA può eseguire solo le operazioni elencate. Inoltre, possono eseguire le operazioni elencate, solo se un'altra istruzione o policy consente l'accesso a tali operazioni.

  La versione `...IfExists` dell'operatore `Bool` garantisce che se la chiave `aws:MultiFactorAuthPresent` manca, la condizione restituisce true. Questo significa che a un utente che accede a un'operazione API con le credenziali di lungo termine, come una chiave di accesso, viene negato l'accesso alle operazioni API non IAM.

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo` e all'istruzione `DenyAllExceptListedIfNoMFA`.

> **Warning**
>
> Non aggiungere l'autorizzazione per l'eliminazione di un dispositivo MFA senza autenticazione MFA. Gli utenti con questa policy potrebbero tentare di autoassegnarsi un dispositivo MFA virtuale e ricevere un errore in cui si specifica che non sono autorizzati a eseguire `iam:DeleteVirtualMFADevice`. In questo caso, non aggiungere tale autorizzazione all'istruzione `DenyAllExceptListedIfNoMFA`. Agli utenti che non si autenticano tramite MFA non deve mai essere consentito di eliminare il dispositivo MFA. Gli utenti potrebbero visualizzare questo errore se hanno in precedenza iniziato ad assegnare un dispositivo MFA virtuale all'utente e annullato il processo. Per risolvere questo problema, tu o un altro amministratore dovete eliminare il dispositivo MFA virtuale esistente dell'utente utilizzando l'API AWS CLI o AWS . Per ulteriori informazioni, consulta Non sono autorizzato a eseguire: iam: MFADevice DeleteVirtual.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": "iam:ListVirtualMFADevices",
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "DenyAllExceptListedIfNoMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:GetUser",
                "iam:ListMFADevices",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice",
                "sts:GetSessionToken"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}
            }
        }
    ]
}
```

AWS: consente agli utenti IAM di modificare la propria password della console nella pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM di modificare la propria AWS Management Console password nella pagina delle credenziali di sicurezza. Questa AWS Management Console pagina mostra le informazioni sull'account e sull'utente, ma l'utente può accedere solo alla propria password. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per informazioni su come gli utenti possono accedere alla pagina delle credenziali di sicurezza, consultaCome gli utenti IAM possono cambiare le proprie password (console).

Che cosa fa questa policy?

- L'istruzione `ViewAccountPasswordRequirements` consente all'utente di visualizzare i requisiti della password dell'account cambiando la propria password utente IAM.
- L'istruzione `ChangeOwnPassword` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).

Questa policy non consente agli utenti di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `ViewAccountPasswordRequirements`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `ChangeOwnPasswords`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "ViewAccountPasswordRequirements",
            "Effect": "Allow",
            "Action": "iam:GetAccountPasswordPolicy",
            "Resource": "*"
        },
        {
            "Sid": "ChangeOwnPassword",
            "Effect": "Allow",
            "Action": [
                "iam:GetUser",
                "iam:ChangePassword"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS: consente agli utenti IAM di gestire la propria password, le chiavi di accesso e le chiavi pubbliche SSH nella pagina Credenziali di sicurezza

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti IAM di gestire la propria password, le chiavi di accesso e i certificati X.509 nella pagina delle credenziali di sicurezza. Questa pagina della AWS Management Console mostra informazioni sull'account, come l'ID account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i dispositivi MFA, i certificati X.509, le chiavi SSH e le credenziali Git. Questa policy di esempio include le autorizzazioni necessarie per visualizzare e modificare solo le password, le chiavi di accesso e il certificato X.509. Per consentire agli utenti di gestire le proprie credenziali con MFA, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza. Per consentire agli utenti di gestire le proprie credenziali senza MFA, consulta AWS: consente agli utenti IAM di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Per informazioni su come gli utenti possono accedere alla pagina delle credenziali di sicurezza, consulta. Come gli utenti IAM possono cambiare le proprie password (console)

Che cosa fa questa policy?

- L'istruzione `AllowViewAccountInfo` consente all'utente di visualizzare le informazioni a livello di account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o

non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`. Questa istruzione include le seguenti operazioni che consentono all'utente di visualizzare informazioni specifiche:

  - `GetAccountPasswordPolicy`: visualizza i requisiti della password dell'account cambiando la propria password utente IAM.
  - `GetAccountSummary`: visualizza l'ID account e l'ID utente canonico dell'account.
- L'istruzione `AllowManageOwnPasswords` consente all'utente di modificare la propria password. Questa istruzione include anche l'operazione `GetUser`, obbligatoria per visualizzare la maggior parte delle informazioni nella pagina My security credentials (Le mie credenziali di sicurezza).
- L'istruzione `AllowManageOwnAccessKeys` consente all'utente di creare, aggiornare ed eliminare le proprie chiavi di accesso. L'utente può anche ottenere informazioni su quando è stata utilizzata l'ultima volta la chiave di accesso specificata.
- L'`AllowManageOwnSSHPublicKeys`istruzione consente all'utente di caricare, aggiornare ed eliminare le proprie chiavi pubbliche SSH per. CodeCommit

Questa policy non consente agli utenti di visualizzare o gestire i propri dispositivi MFA. Inoltre, non sono in grado di visualizzare la pagina Utenti nella console IAM o utilizzare questa pagina per accedere alle proprie informazioni utente. Per consentire questa operazione, aggiungere l'operazione `iam:ListUsers` all'istruzione `AllowViewAccountInfo`. Inoltre, non consente agli utenti di cambiare la password sulla proprio pagina utente. Per consentire questa operazione, aggiungi le operazioni `iam:GetLoginProfile` e `iam:UpdateLoginProfile` all'istruzione `AllowManageOwnPasswords`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy",
                "iam:GetAccountSummary"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswords",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:ChangePassword",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:GetAccessKeyLastUsed"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnSSHPublicKeys",
            "Effect": "Allow",
            "Action": [
                "iam:DeleteSSHPublicKey",
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:UpdateSSHPublicKey",
                "iam:UploadSSHPublicKey"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

AWS: nega l'accesso in AWS base alla regione richiesta

Questo esempio illustra come creare una policy basata sull'identità che neghi l'accesso a qualsiasi operazione esterna alle regioni specificate utilizzando la chiave di condizione `aws:RequestedRegion`, fatta eccezione per le operazioni nei servizi specificati tramite `NotAction`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy utilizza l'elemento `NotAction` con l'effetto `Deny`, che rifiuta esplicitamente l'accesso a tutte le operazioni che non sono elencate nella dichiarazione. Le azioni su IAM CloudFront, Route 53 e AWS Support sui servizi non devono essere negate, perché si tratta di servizi AWS globali molto diffusi con un unico endpoint che si trova fisicamente nella `us-east-1` regione. Poiché tutte le richieste a questi servizi vengono effettuate alla regione `us-east-1`, le richieste vengono rifiutate senza l'elemento `NotAction`. Modifica questo elemento per includere operazioni per altri servizi globali AWS che utilizzi, ad esempio `budgets`, `globalaccelerator`, `importexport`, `organizations` o `waf`. Alcuni altri servizi globali, come AWS Chatbot and AWS Device Farm, sono servizi globali con endpoint che si trovano fisicamente nella `us-west-2` regione. Per ulteriori informazioni su tutti i servizi che dispongono di un singolo endpoint globale, consulta Regioni ed endpoint AWS nella Riferimenti generali di AWS. Per ulteriori informazioni sull'utilizzo dell'elemento `NotAction` con l'effetto `Deny`, consulta Elementi delle policy JSON IAM: NotAction.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAllOutsideRequestedRegions",
            "Effect": "Deny",
            "NotAction": [
                "cloudfront:*",
                "iam:*",
                "route53:*",
                "support:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "eu-central-1",
                        "eu-west-1",
                        "eu-west-2",
                        "eu-west-3"
                    ]
```

```
                }
            }
        }
    ]
}
```

AWS: nega l'accesso in AWS base all'IP di origine

Questo esempio mostra come è possibile creare una policy basata sull'identità che neghi l'accesso a tutte le AWS azioni dell'account quando la richiesta proviene da soggetti esterni all'intervallo IP specificato. La policy è utile quando gli indirizzi IP per la tua azienda sono all'interno di determinati intervalli. In questo esempio, la richiesta verrà rifiutata a meno che non provenga dall'intervallo CIDR 192.0.2.0/24 o 203.0.113.0/24. La politica non nega le richieste effettuate dai AWS servizi che utilizzano l'indirizzo IP Inoltro delle sessioni di accesso del richiedente originale.

Fare attenzione a utilizzare condizioni negative nella stessa dichiarazione di policy come `"Effect": "Deny"`. In questo caso, le operazioni specificate nella dichiarazione di policy vengono negate in modo esplicito in tutte le condizioni, ad eccezione di quelle specificate.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

Quando altre policy consentono operazioni, le entità possono effettuare richieste all'interno dell'intervallo di indirizzi IP. Un AWS servizio può anche effettuare richieste utilizzando le credenziali del principale. Quando un'entità effettua una richiesta al di fuori dell'intervallo IP, la richiesta viene negata.

Per ulteriori informazioni sull'utilizzo della chiave di condizione `aws:SourceIp`, incluse le informazioni su quando `aws:SourceIp` potrebbe non funzionare nelle policy, consulta AWS chiavi di contesto della condizione globale.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
```

```
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            }
        }
    }
}
```

AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange

Questo esempio mostra come potresti creare una politica basata sull'identità che neghi l'accesso a tutte le risorse AWS che non appartengono al tuo account, ad eccezione delle risorse necessarie per il normale funzionamento. AWS Data Exchange Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

È possibile creare una politica simile per limitare l'accesso alle risorse all'interno di un'organizzazione o di un'unità organizzativa, contabilizzando al contempo le risorse di AWS Data Exchange proprietà utilizzando i tasti di condizione e. `aws:ResourceOrgPaths` `aws:ResourceOrgID`

Se lo utilizzi AWS Data Exchange nel tuo ambiente, il servizio crea e interagisce con risorse come i bucket Amazon S3 di proprietà dell'account del servizio. Ad esempio, AWS Data Exchange invia richieste ai bucket Amazon S3 di proprietà del AWS Data Exchange servizio per conto del principale IAM (utente o ruolo) che richiama le API. AWS Data Exchange In tal caso, l'utilizzo `aws:ResourceAccount`aws:ResourceOrgPaths, o `aws:ResourceOrgID` nell'ambito di una policy, senza tenere conto delle risorse di AWS Data Exchange proprietà, nega l'accesso ai bucket di proprietà dell'account di servizio.

- L'istruzione `DenyAllAwsResourcesOutsideAccountExceptS3` utilizza l'elemento `NotAction` con l'effetto Deny che nega esplicitamente l'accesso a tutte le operazioni non elencate nell'istruzione e che non appartengono all'account elencato. L'elemento `NotAction` indica le eccezioni a questa istruzione. Queste azioni fanno eccezione a questa dichiarazione perché se le azioni vengono eseguite su risorse create da AWS Data Exchange, la policy le nega.
- L'istruzione `DenyAllS3ResoucesOutsideAccountExceptDataExchange` utilizza una combinazione delle condizioni `ResourceAccount` e `CalledVia` per negare l'accesso

alle tre operazioni di Amazon S3 escluse nell'istruzione precedente. L'istruzione nega le operazioni se le risorse non appartengono all'account elencato e se il servizio chiamante non è AWS Data Exchange. L'istruzione non nega le operazioni se la risorsa appartiene all'account elencato o l'operazione viene eseguita dal principale del servizio elencato, `dataexchange.amazonaws.com`.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza l'effetto `Deny`, che neghi esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

L'esempio seguente mostra come configurare la policy per consentire l'accesso ai bucket Amazon S3 richiesti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllAwsReourcesOutsideAccountExceptAmazonS3",
      "Effect": "Deny",
      "NotAction": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        }
      }
    },
    {
      "Sid": "DenyAllS3ResourcesOutsideAccountExceptDataExchange",
      "Effect": "Deny",
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        },
        "ForAllValues:StringNotEquals": {
          "aws:CalledVia": [
            "dataexchange.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

AWS Data Pipeline: nega l'accesso alle DataPipeline pipeline che un utente non ha creato

Questo esempio mostra come creare una policy basata sull'identità che neghi l'accesso alle pipeline che non sono state create da un utente. Se il valore del campo `PipelineCreator` corrisponde al nome dell'utente IAM, le operazioni specificate non saranno rifiutate. Questa policy concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "ExplicitDenyIfNotTheOwner",
            "Effect": "Deny",
            "Action": [
                "datapipeline:ActivatePipeline",
                "datapipeline:AddTags",
                "datapipeline:DeactivatePipeline",
                "datapipeline:DeletePipeline",
                "datapipeline:DescribeObjects",
                "datapipeline:EvaluateExpression",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:PollForTask",
                "datapipeline:PutPipelineDefinition",
                "datapipeline:QueryObjects",
                "datapipeline:RemoveTags",
                "datapipeline:ReportTaskProgress",
                "datapipeline:ReportTaskRunnerHeartbeat",
                "datapipeline:SetStatus",
                "datapipeline:SetTaskStatus",
                "datapipeline:ValidatePipelineDefinition"
            ],
            "Resource": ["*"],
            "Condition": {
                "StringNotEquals": {"datapipeline:PipelineCreator": "${aws:userid}"}
            }
        }
    ]
}
```

Amazon DynamoDB: consente l'accesso a una tabella specifica

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo alla tabella `MyTable` di DynamoDB. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Important**
>
> Questa policy consente tutte le operazioni che possono essere eseguite su una tabella DynamoDB. Per esaminare queste operazioni, consulta Autorizzazioni API di DynamoDB: riferimento a operazioni, risorse e condizioni nella Guida per gli sviluppatori di Amazon DynamoDB. Puoi fornire le stesse autorizzazioni elencando ogni singola azione. Tuttavia, se

utilizzi il carattere jolly (*) nell'elemento `Action`, ad esempio `"dynamodb:List*"`, non sarà necessario aggiornare la policy se DynamoDB aggiunge una nuova operazione Elenco.

Questa policy consente le operazioni solo sulle tabelle DynamoDB con il nome specificato. Per consentire ai tuoi utenti di `Read` accedere a tutto ciò che è in DynamoDB, puoi anche allegare AmazonDynamola policy gestita dal ReadOnlyAccess AWS DB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListAndDescribe",
            "Effect": "Allow",
            "Action": [
                "dynamodb:List*",
                "dynamodb:DescribeReservedCapacity*",
                "dynamodb:DescribeLimits",
                "dynamodb:DescribeTimeToLive"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SpecificTable",
            "Effect": "Allow",
            "Action": [
                "dynamodb:BatchGet*",
                "dynamodb:DescribeStream",
                "dynamodb:DescribeTable",
                "dynamodb:Get*",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:BatchWrite*",
                "dynamodb:CreateTable",
                "dynamodb:Delete*",
                "dynamodb:Update*",
                "dynamodb:PutItem"
            ],
            "Resource": "arn:aws:dynamodb:*:*:table/MyTable"
        }
    ]
}
```

Amazon DynamoDB: consente l'accesso ad attributi specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso ad attributi DynamoDB specifici. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Il requisito `dynamodb:Select` impedisce all'operazione API di restituire qualsiasi attributo per cui non si abbia il permesso, come ad esempio da una proiezione di indice. Per ulteriori informazioni sulle chiavi di condizione DynamoDB, consulta Specifica delle condizioni: uso delle chiavi di condizione nella Guida per gli sviluppatori di Amazon DynamoDB. Per ulteriori informazioni sulle condizioni multiple o sulle chiavi di condizioni multiple all'interno di un blocco di policy IAM `Condition`, consultare la pagina Valori multipli in una condizione

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetItem",
                "dynamodb:BatchGetItem",
                "dynamodb:Query",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteItem",
                "dynamodb:BatchWriteItem"
            ],
            "Resource": ["arn:aws:dynamodb:*:*:table/table-name"],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:Attributes": [
                        "column-name-1",
                        "column-name-2",
                        "column-name-3"
                    ]
                },
                "StringEqualsIfExists": {"dynamodb:Select": "SPECIFIC_ATTRIBUTES"}
            }
        }
    ]
```

```
}
```

Amazon DynamoDB: consente l'accesso a livello di elemento a DynamoDB in base a un ID Amazon Cognito

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso a livello di elemento alla tabella DynamoDB `MyTable` in base all'ID utente di un pool di identità Amazon Cognito. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per utilizzare questa policy, devi strutturare la tabella DynamoDB in modo che l'ID utente del pool di identità Amazon Cognito costituisca la chiave di partizione. Per ulteriori informazioni, consulta Creazione di una tabella nella Guida per gli sviluppatori di Amazon DynamoDB.

Per ulteriori informazioni sulle chiavi di condizione DynamoDB, consulta Specifica delle condizioni: uso delle chiavi di condizione nella Guida per gli sviluppatori di Amazon DynamoDB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:UpdateItem"
            ],
            "Resource": ["arn:aws:dynamodb:*:*:table/MyTable"],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": ["${cognito-identity.amazonaws.com:sub}"]
                }
            }
        }
    ]
}
```

Amazon EC2: Collegare o distaccare volumi Amazon EBS alle istanze EC2 in base ai tag

Questo esempio mostra come creare una policy basata sull'identità che consenta ai proprietari di volumi EBS di collegare o scollegare i propri volumi EBS, definiti utilizzando il tag `VolumeUser`, alle istanze EC2 contrassegnate con tag come istanze di sviluppo (`Department=Development`). Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per ulteriori informazioni sulla creazione di policy IAM per controllare l'accesso alle risorse Amazon EC2, consulta Controlling Access to Amazon EC2 Resources nella Amazon EC2 User Guide.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Department": "Development"}
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/VolumeUser": "${aws:username}"}
            }
        }
    ]
}
```

Amazon EC2: consente l'avvio di istanze EC2 in una sottorete specifica, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di elencare informazioni per tutti gli oggetti EC2 e di avviare istanze EC2 in una specifica sottorete. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:GetConsole*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/subnet-subnet-id",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Amazon EC2: consente la gestione dei gruppi di sicurezza EC2 con una specifica coppia chiave-valore tag, in modo programmatico e nella console

In questo esempio viene illustrato come creare una policy basata sull'identità che conceda agli utenti l'autorizzazione a intraprendere determinate operazioni per i gruppi di sicurezza con lo stesso tag.

Questa policy concede le autorizzazioni per visualizzare i gruppi di sicurezza nella console Amazon EC2, per aggiungere e rimuovere le regole in entrata e in uscita e per elencare e modificare le descrizioni delle regole per i gruppi di sicurezza esistenti con il tag `Department=Test`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups",
         "ec2:DescribeSecurityGroupRules",
         "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
      ],
      "Resource": [
         "arn:aws:ec2:region:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/Department": "Test"
         }
      }
     },
     {
      "Effect": "Allow",
      "Action": [
```

```
            "ec2:ModifySecurityGroupRules"
         ],
         "Resource": [
            "arn:aws:ec2:region:111122223333:security-group-rule/*"
         ]
       }
    ]
}
```

Amazon EC2: consente l'avvio o l'arresto di istanze EC2 che un utente ha contrassegnato, a livello programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente IAM di avviare o arrestare le istanze EC2, ma solo se il `Owner` del tag dell'istanza ha il valore del nome utente di quell'utente. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

EC2: avvio o arresto di istanze in base ai tag

Questo esempio mostra come creare una policy basata sull'identità che consenta di avviare o arrestare le istanze con la coppia chiave-valore di tag `Project = DataAnalytics`, ma solo ai principali con la coppia chiave-valore di tag `Department = Data`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La condizione nella policy restituisce true se entrambe le parti della condizione sono vere. L'istanza deve avere il tag `Project=DataAnalytics`. Inoltre, il principale IAM (utente o ruolo) da cui proviene la richiesta deve avere il tag `Department=Data`.

> **Note**
>
> Come best practice, collega policy con la chiave di condizione `aws:PrincipalTag` a gruppi IAM, nel caso in cui non tutti gli utenti dispongano del tag specificato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "StartStopIfTags",
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:region:account-id:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Project": "DataAnalytics",
                    "aws:PrincipalTag/Department": "Data"
                }
            }
        }
    ]
}
```

EC2: avvio o arresto di istanze in base alla corrispondenza dei tag della risorsa e del principale

Questo esempio mostra come creare una policy basata sull'identità che consenta a un principale di avviare o interrompere un'istanza Amazon EC2 quando il tag della risorsa dell'istanza e il tag del principale hanno lo stesso valore per la chiave di tag `CostCenter`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Note**
>
> Come best practice, collega policy con la chiave di condizione `aws:PrincipalTag` a gruppi IAM, nel caso in cui non tutti gli utenti dispongano del tag specificato.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "ec2:startInstances",
            "ec2:stopInstances"
        ],
        "Resource": "*",
        "Condition": {"StringEquals":
            {"aws:ResourceTag/CostCenter": "${aws:PrincipalTag/CostCenter}"}}
    }
}
```

Amazon EC2: consente l'accesso completo a EC2 entro una regione specifica, a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a EC2 in una regione specifica. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in

Creazione di una policy o Modifica di una policy. Per un elenco dei codici di regione, consulta Regioni disponibili nella Guida per l'utente di Amazon EC2.

In alternativa, puoi utilizzare la chiave di condizione `aws:RequestedRegion`, supportata da tutte le operazioni API di Amazon EC2. Per ulteriori informazioni, consulta Esempio: limitazione dell'accesso a una regione specifica nella Guida per l'utente di Amazon EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ec2:*",
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2"
                }
            }
        }
    ]
}
```

Amazon EC2: consente l'avvio o l'arresto di un'istanza EC2 e la modifica di un gruppo di sicurezza, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'avvio o l'arresto di un'istanza EC2 specifica e la modifica di un determinato gruppo di sicurezza. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSecurityGroupReferences",
        "ec2:DescribeStaleSecurityGroups"
```

```
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/i-instance-id",
        "arn:aws:ec2:*:*:security-group/sg-security-group-id"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Amazon EC2: richiede MFA (GetSessionToken) per operazioni EC2 specifiche

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a tutte le operazioni AWS API in Amazon EC2. Tuttavia, rifiuta esplicitamente l'accesso alle operazioni API `StopInstances` e `TerminateInstances` se l'utente non viene autenticato utilizzando l'autenticazione a più fattori (MFA, Multi-Factor Authentication). Per eseguire questa operazione a livello di programmazione, l'utente deve includere valori `SerialNumber` e `TokenCode` opzionali durante la chiamata all'operazione `GetSessionToken`. Questa operazione restituisce credenziali temporanee autenticate utilizzando MFA. Per ulteriori informazioni, consulta. GetSessionToken GetSessionToken: le credenziali temporanee per gli utenti in ambienti non attendibili

Che cosa fa questa policy?

- L'istruzione `AllowAllActionsForEC2` consente tutte le operazioni Amazon EC2.
- La dichiarazione `DenyStopAndTerminateWhenMFAIsNotPresent` rifiuta le operazioni `TerminateInstances` e `StopInstances` quando manca il contesto MFA. Ciò significa che le operazioni vengono rifiutate quando manca il contesto dell'autenticazione a più fattori (ovvero, MFA non è stato utilizzato). Un rifiuto sostituisce il consenso.

> **Note**
>
> Il controllo della condizione per `MultiFactorAuthPresent` nella dichiarazione `Deny` non deve essere `{"Bool":{"aws:MultiFactorAuthPresent":false}}`, perché tale chiave non è presente e non può essere valutata quando MFA non viene utilizzato. Utilizzare invece il controllo `BoolIfExists` per vedere se la chiave è presente prima di controllare il valore. Per ulteriori informazioni, consulta ... IfExists operatori di condizionamento.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllActionsForEC2",
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*"
        },
        {
            "Sid": "DenyStopAndTerminateWhenMFAIsNotPresent",
            "Effect": "Deny",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {"aws:MultiFactorAuthPresent": false}
            }
        }
    ]
}
```

Amazon EC2: limita la chiusura delle istanze EC2 a un intervallo di indirizzi IP

Questo esempio mostra come creare una policy basata sull'identità che limiti le istanze EC2, consentendo l'operazione, ma negando esplicitamente l'accesso quando la richiesta proviene da un indirizzo IP esterno all'intervallo specificato. La policy è utile quando gli indirizzi IP per la tua azienda sono all'interno di determinati intervalli. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy,

sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se questa politica viene utilizzata in combinazione con altre politiche che consentono l'`ec2:TerminateInstances`azione (come la politica FullAccess AWS gestita da AmazonEC2), l'accesso viene negato. Questo perché una dichiarazione di rifiuto esplicita prevale su una dichiarazione di consenso. Per ulteriori informazioni, consulta the section called “Determinazione se una richiesta è consentita o rifiutata in un account”.

> **Important**
>
> La chiave `aws:SourceIp` condizionale nega l'accesso a un AWS servizio, ad esempio AWS CloudFormation, che effettua chiamate per tuo conto. Per ulteriori informazioni su come utilizzare la chiave di condizione `aws:SourceIp`, consultare AWS chiavi di contesto della condizione globale.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

IAM: accesso all'API del simulatore di policy

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo dell'API del simulatore di policy per le policy collegate a un utente, un gruppo o un ruolo nell' Account AWS corrente. Questa policy consente inoltre l'accesso per simulare le policy meno sensibili passate all'API come stringhe. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetContextKeysForCustomPolicy",
                "iam:GetContextKeysForPrincipalPolicy",
                "iam:SimulateCustomPolicy",
                "iam:SimulatePrincipalPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Per consentire a un utente di accedere alla console del simulatore di policy per simulare le policy associate a un utente, gruppo o ruolo nel sistema corrente, consulta. Account AWSIAM: accesso alla console del simulatore di policy

IAM: accesso alla console del simulatore di policy

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo della console del simulatore di policy per le policy collegate a un utente, un gruppo o un ruolo nell' Account AWS corrente. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

È possibile accedere alla console del simulatore di policy IAM all'indirizzo https://policysim.aws.amazon.com/

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListAttachedRolePolicies",
                "iam:ListAttachedUserPolicies",
                "iam:ListGroups",
                "iam:ListGroupPolicies",
                "iam:ListGroupsForUser",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListUserPolicies",
                "iam:ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

IAM: assumere ruoli che dispongono di un tag specifico

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente IAM di assumere ruoli con la coppia chiave-valore di tag `Project = ExampleCorpABC`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se un ruolo con questo tag esiste nello stesso account dell'utente, l'utente può assumere tale ruolo. Se un ruolo con questo tag esiste in un account diverso da quello dell'utente, sono richieste autorizzazioni aggiuntive. La policy di attendibilità del ruolo tra account deve anche consentire

all'utente o a tutti i membri dell'account dell'utente di assumere il ruolo. Per ulteriori informazioni sull'utilizzo di ruoli per l'accesso tra account, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeTaggedRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:ResourceTag/Project": "ExampleCorpABC"}
            }
        }
    ]
}
```

IAM: consente e rifiuta l'accesso a più servizi a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo a diversi servizi e l'accesso autonomo limitato in IAM. Rifiuta inoltre l'accesso al bucket `logs` di Amazon S3 o all'istanza `i-1234567890abcdef0` Amazon EC2. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Warning**
>
> Questa policy consente l'accesso completo a tutte le operazioni e risorse in più servizi. Questa policy deve essere applicata solo ad amministratori fidati.

Puoi usare questa policy come un limite delle autorizzazioni per definire il numero massimo di autorizzazioni che una policy basata su identità può concedere a un utente IAM. Per ulteriori informazioni, consulta Delega di responsabilità ad altri mediante i limiti delle autorizzazioni. Quando la policy viene usata come un limite delle autorizzazioni per un utente, le dichiarazioni definiscono i seguenti limiti:

- L'istruzione `AllowServices` consente l'accesso completo ai servizi AWS specificati. Ciò significa che le operazioni dell'utente in questi servizi sono limitate solo dalle policy di autorizzazioni collegate all'utente.
- La dichiarazione `AllowIAMConsoleForCredentials` consente l'accesso per elencare tutti gli utenti IAM. Questo accesso è necessario per navigare nella pagina Users (Utenti) nella AWS Management Console. Inoltre, consente di visualizzare i requisiti associati alle password per l'account, operazione necessaria per permettere all'utente di modificare la sua password.
- L'istruzione `AllowManageOwnPasswordAndAccessKeys` consente agli utenti di gestire solo le proprie chiavi di accesso programmatiche e password della console. Questo è importante perché se un'altra policy offre a un utente l'accesso IAM completo, tale utente può modificare le sue autorizzazioni o quelle di altri utenti. Questa istruzione impedisce che ciò si verifichi.
- L'istruzione `DenyS3Logs` nega esplicitamente l'accesso al bucket `logs`. Questa policy applica limitazioni aziendali all'utente.
- L'istruzione `DenyEC2Production` nega esplicitamente l'accesso all'istanza `i-1234567890abcdef0`.

Questa policy non consente l'accesso ad altri servizi o operazioni. Quando la politica viene utilizzata come limite di autorizzazioni per un utente, anche se altre politiche allegate all'utente consentono tali azioni, AWS nega la richiesta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServices",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "cloudwatch:*",
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIAMConsoleForCredentials",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnPasswordAndAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*LoginProfile*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::logs",
                "arn:aws:s3:::logs/*"
            ]
        },
        {
            "Sid": "DenyEC2Production",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "arn:aws:ec2:*:*:instance/i-1234567890abcdef0"
        }
    ]
}
```

IAM: aggiunta di un tag specifico a un utente con un determinato tag

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere la chiave di tag `Department` con i valori di tag `Marketing`, `Development` o `QualityAssurance` a un utente IAM. Tale utente deve già includere la coppia chiave-valore di tag `JobFunction = manager`. È possibile usare questa policy per richiedere che un responsabile appartenga solo a uno dei tre reparti. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'istruzione `ListTagsForAllUsers` consente di visualizzare i tag per tutti gli utenti nell'account.

La prima condizione nell'istruzione `TagManagerWithSpecificDepartment` utilizza l'operatore di condizione `StringEquals`. La condizione restituisce true se entrambe le parti della condizione sono vere. L'utente che necessita di tag deve già disporre del tag `JobFunction=Manager`. La richiesta deve includere la chiave di tag `Department` con uno dei valori di tag elencati.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se tutte le chiavi di tag nella richiesta corrispondono alla chiave nella policy. Ciò significa che l'unica chiave di tag nella richiesta deve essere `Department`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Chiavi di contesto multivalore.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListTagsForAllUsers",
            "Effect": "Allow",
            "Action": [
                "iam:ListUserTags",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagManagerWithSpecificDepartment",
            "Effect": "Allow",
            "Action": "iam:TagUser",
            "Resource": "*",
            "Condition": {"StringEquals": {
                "iam:ResourceTag/JobFunction": "Manager",
                "aws:RequestTag/Department": [
                    "Marketing",
                    "Development",
                    "QualityAssurance"
                    ]
                },
                "ForAllValues:StringEquals": {"aws:TagKeys": "Department"}
            }
        }
    ]
}
```

IAM: aggiunta di un determinato tag con valori specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere solo la chiave di tag `CostCenter` e il valore di tag `A-123` o il valore di tag `B-456` a qualsiasi ruolo o utente IAM. Puoi utilizzare questa policy per limitare il tagging a una chiave di tag e a un set di valori di tag specifici. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'istruzione `ConsoleDisplay` consente di visualizzare i tag per tutti gli utenti e i ruoli nell'account.

La prima condizione nell'istruzione `AddTag` utilizza l'operatore di condizione `StringEquals`. La condizione restituisce true se la richiesta include la chiave di tag `CostCenter` con uno dei valori di tag elencati.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se tutte le chiavi di tag nella richiesta corrispondono alla chiave nella policy. Ciò significa che l'unica chiave di tag nella richiesta deve essere `CostCenter`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Chiavi di contesto multivalore.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConsoleDisplay",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:GetUser",
                "iam:ListRoles",
                "iam:ListRoleTags",
                "iam:ListUsers",
                "iam:ListUserTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AddTag",
            "Effect": "Allow",
            "Action": [
                "iam:TagUser",
```

```
                "iam:TagRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CostCenter": [
                        "A-123",
                        "B-456"
                    ]
                },
                "ForAllValues:StringEquals": {"aws:TagKeys": "CostCenter"}
            }
        }
    ]
}
```

IAM: creazione di nuovi utenti solo con tag specifici

Questo esempio mostra come creare una policy basata sull'identità che consenta la creazione di utenti IAM, ma solo con una o entrambe le chiavi di tag `Department` e `JobFunction`. La chiave di tag `Department` deve avere il valore di tag `Development` o `QualityAssurance`. La chiave di tag `JobFunction` deve avere il valore di tag `Employee`. È possibile usare questa policy per richiedere che i nuovi utenti dispongano di una mansione e un reparto specifici. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La prima condizione nell'istruzione utilizza l'operatore di condizione `StringEqualsIfExists`. Se un tag con la chiave `Department` o `JobFunction` è presente nella richiesta, il tag deve avere il valore specificato. Se non è presente alcuna chiave, questa condizione viene valutata come true. La condizione viene valutata come false solo se una delle chiavi di condizione specificate è presente nella richiesta, ma ha un valore diverso da quelli consentiti. Per ulteriori informazioni sull'utilizzo di `IfExists`, consultare ... IfExists operatori di condizionamento.

La seconda condizione utilizza l'operatore di condizione `ForAllValues:StringEquals`. La condizione restituisce true se si verifica una corrispondenza tra ognuna delle chiavi di tag specificate nella richiesta e almeno un valore nella policy. Ciò significa che tutti i tag nella richiesta devono essere in questo elenco. Tuttavia, la richiesta può includere solo uno dei tag nell'elenco. Ad esempio, puoi creare un utente IAM con il solo tag `Department=QualityAssurance`. Tuttavia, non puoi

creare un utente IAM con il tag `JobFunction=employee` e il tag `Project=core`. Per ulteriori informazioni sull'utilizzo di `ForAllValues`, consultare Chiavi di contesto multivalore.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagUsersWithOnlyTheseTags",
            "Effect": "Allow",
            "Action": [
                "iam:CreateUser",
                "iam:TagUser"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "aws:RequestTag/Department": [
                        "Development",
                        "QualityAssurance"
                    ],
                    "aws:RequestTag/JobFunction": "Employee"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "Department",
                        "JobFunction"
                    ]
                }
            }
        }
    ]
}
```

IAM: generazione e recupero di report di credenziali IAM

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta agli utenti di generare e scaricare un report che elenca tutti gli utenti IAM del loro gruppo. Account AWS Il report include lo stato delle credenziali dell'utente, incluse le password, le chiavi di accesso, i dispositivi MFA e i certificati di firma. Questa policy concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

Per ulteriori informazioni sui report delle credenziali, consultare la pagina Recupero dei report delle credenziali per l' Account AWS.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateCredentialReport",
            "iam:GetCredentialReport"
        ],
        "Resource": "*"
    }
}
```

IAM: consente di gestire l'appartenenza di un gruppo a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiornare l'appartenenza al gruppo denominato `MarketingTeam`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Che cosa fa questa policy?

- La `ViewGroups` dichiarazione consente all'utente di elencare tutti gli utenti e i gruppi nella AWS Management Console. Inoltre, consente all'utente di visualizzare informazioni di base sugli utenti nell'account. Queste autorizzazioni devono essere nella propria istruzione perché non supportano o non devono specificare l'ARN di una risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`.
- La dichiarazione `ViewEditThisGroup` consente all'utente di visualizzare le informazioni sul gruppo `MarketingTeam` e aggiungere o rimuovere utenti da tale gruppo.

Questa policy non consente all'utente di visualizzare o modificare le autorizzazioni degli utenti o del gruppo `MarketingTeam`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "ViewGroups",
            "Effect": "Allow",
            "Action": [
                "iam:ListGroups",
                "iam:ListUsers",
                "iam:GetUser",
                "iam:ListGroupsForUser"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewEditThisGroup",
            "Effect": "Allow",
            "Action": [
                "iam:AddUserToGroup",
                "iam:RemoveUserFromGroup",
                "iam:GetGroup"
            ],
            "Resource": "arn:aws:iam::*:group/MarketingTeam"
        }
    ]
}
```

IAM: gestione di un tag specifico

Questo esempio mostra come creare una policy basata sull'identità che consenta di aggiungere e rimuovere il tag IAM con la chiave di tag `Department` dalle entità IAM (utenti e ruoli). Questa policy non limita il valore del tag `Department`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:TagUser",
            "iam:TagRole",
            "iam:UntagUser",
```

```
            "iam:UntagRole"

        ],
        "Resource": "*",
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys": "Department"}}
    }
}
```

IAM: passa un ruolo IAM a un AWS servizio specifico

Questo esempio mostra come è possibile creare una policy basata sull'identità che consenta di passare qualsiasi ruolo del servizio IAM al servizio Amazon. CloudWatch Questa politica concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Un ruolo di servizio è un ruolo IAM che specifica un AWS servizio come principale che può assumere il ruolo. Questo consente al servizio di assumere il ruolo e accedere a risorse in altri servizi a tuo nome. Per consentire CloudWatch ad Amazon di assumere il ruolo che conferisci, devi specificare il responsabile del `cloudwatch.amazonaws.com` servizio come responsabile nella politica di fiducia del tuo ruolo. Il principale del servizio è definito dal servizio. Per ulteriori informazioni sul principale del servizio per un servizio, consultare la documentazione per quel servizio. Per alcuni servizi, consulta AWS servizi che funzionano con IAM e cerca i servizi che hanno Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio. Cerca `amazonaws.com` per visualizzare il principale del servizio.

Per ulteriori informazioni sul passaggio di un ruolo del servizio al servizio, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:PassedToService": "cloudwatch.amazonaws.com"}
```

```
            }
        }
    ]
}
```

IAM: consente l'accesso in sola lettura alla console IAM senza la creazione di report

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di eseguire qualsiasi operazione IAM che inizia con la stringa `Get` o `List`. Quando gli utenti utilizzano la console, la console effettua richieste a IAM per elencare gruppi, utenti, ruoli e policy e generare report su tali risorse.

L'asterisco funge da carattere jolly. Quando utilizzi `iam:Get*` in una policy, le autorizzazioni risultanti includono tutte le operazioni IAM che iniziano con `Get`, ad esempio `GetUser` e `GetRole`. I caratteri jolly sono utili quando nuovi tipi di entità vengono aggiunti a IAM in futuro. In tal caso, le autorizzazioni concesse dalla policy consentono automaticamente all'utente di elencare e ottenere i dettagli su queste nuove entità.

Questa policy non può essere utilizzata per generare report o dettagli dell'ultimo accesso al servizio. Per una policy diversa che lo consenta, consulta IAM: consente l'accesso in sola lettura alla console IAM.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    }
}
```

IAM: consente l'accesso in sola lettura alla console IAM

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di eseguire qualsiasi operazione IAM che inizia con la stringa `Get`, `List` o `Generate`. Quando gli utenti utilizzano la console IAM, la console effettua richieste per elencare gruppi, utenti, ruoli e policy e generare report su tali risorse.

L'asterisco funge da carattere jolly. Quando utilizzi `iam:Get*` in una policy, le autorizzazioni risultanti includono tutte le operazioni IAM che iniziano con `Get`, ad esempio `GetUser` e `GetRole`. L'uso di un carattere jolly è utile, in particolare se in futuro vengono aggiunti nuovi tipi di entità a IAM. In tal caso, le autorizzazioni concesse dalla policy consentono automaticamente all'utente di elencare e ottenere i dettagli su queste nuove entità.

Utilizza questa policy per l'accesso alla console che include le autorizzazioni per generare report o i dettagli dell'ultimo accesso al servizio. Per una policy diversa che non consenta la generazione di operazioni, consulta IAM: consente l'accesso in sola lettura alla console IAM senza la creazione di report.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:Get*",
            "iam:List*",
            "iam:Generate*"
        ],
        "Resource": "*"
    }
}
```

IAM: consente a utenti IAM specifici di gestire un gruppo a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a specifici utenti IAM di gestire il gruppo `AllUsers`. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Che cosa fa questa policy?

- L'istruzione `AllowAllUsersToListAllGroups` consente di elencare tutti i gruppi. Questa operazione è necessaria per l'accesso alla console. Questa autorizzazione deve trovarsi nella propria dichiarazione perché non supporta un ARN della risorsa. Le autorizzazioni specificano invece `"Resource" : "*"`.

- L'istruzione `AllowAllUsersToViewAndManageThisGroup` consente tutte le operazioni del gruppo che possono essere eseguite sul tipo di risorsa del gruppo. Non consente l'operazione `ListGroupsForUser`, che può essere eseguita su un tipo di risorsa dell'utente e non un tipo di risorsa del gruppo. Per ulteriori informazioni sui tipi di risorsa che è possibile specificare per un'operazione IAM, consulta Operazioni, risorse e chiavi di condizione per AWS Identity and Access Management.
- L'istruzione `LimitGroupManagementAccessToSpecificUsers` nega agli utenti con i nomi specificati l'accesso in scrittura e le operazioni del gruppo di gestione delle autorizzazioni. Quando un utente specificato nella policy tenta di apportare modifiche al gruppo, questa istruzione non rifiuta la richiesta. Questa richiesta è consentita dall'istruzione `AllowAllUsersToViewAndManageThisGroup`. Se altri utenti tentano di eseguire queste operazioni, la richiesta viene rifiutata. Puoi visualizzare le operazioni IAM che vengono definite con i livelli di accesso Scrittura o Gestione delle autorizzazioni durante la creazione di questa policy nella console IAM. A tale scopo, passare dalla scheda JSON alla scheda Visual editor. Per ulteriori informazioni sui livelli di accesso. consulta Operazioni, risorse e chiavi di condizione per AWS Identity and Access Management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllUsersToListAllGroups",
            "Effect": "Allow",
            "Action": "iam:ListGroups",
            "Resource": "*"
        },
        {
            "Sid": "AllowAllUsersToViewAndManageThisGroup",
            "Effect": "Allow",
            "Action": "iam:*Group*",
            "Resource": "arn:aws:iam::*:group/AllUsers"
        },
        {
            "Sid": "LimitGroupManagementAccessToSpecificUsers",
            "Effect": "Deny",
            "Action": [
                "iam:AddUserToGroup",
                "iam:CreateGroup",
                "iam:RemoveUserFromGroup",
```

```
                "iam:DeleteGroup",
                "iam:AttachGroupPolicy",
                "iam:UpdateGroup",
                "iam:DetachGroupPolicy",
                "iam:DeleteGroupPolicy",
                "iam:PutGroupPolicy"
            ],
            "Resource": "arn:aws:iam::*:group/AllUsers",
            "Condition": {
                "StringNotEquals": {
                    "aws:username": [
                        "srodriguez",
                        "mjackson",
                        "adesai"
                    ]
                }
            }
        }
    ]
}
```

IAM: consente l'impostazione dei requisiti della password dell'account a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta a un utente di visualizzare e aggiornare i requisiti della password dell'account. I requisiti della password specificano i requisiti di complessità e i periodi di rotazione obbligatori per le password dei membri dell'account. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

Per Scopri come impostare la policy dei requisiti della password dell'account per l'account, consulta Impostazione di una policy delle password dell'account per utenti IAM.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GetAccountPasswordPolicy",
            "iam:UpdateAccountPasswordPolicy"
        ],
        "Resource": "*"
    }
```

```
}
```

IAM: accesso all'API simulatore di policy basata sul percorso degli utenti

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo dell'API del simulatore di policy solo da parte degli utenti con il percorso `Department/Development`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetContextKeysForPrincipalPolicy",
                "iam:SimulatePrincipalPolicy"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:user/Department/Development/*"
        }
    ]
}
```

> **Note**
>
> Per creare una policy che consente di utilizzare la console del simulatore di policy per gli utenti con percorso `Department/Development`, consultare IAM: accesso alla console del simulatore di policy in base al percorso utente.

IAM: accesso alla console del simulatore di policy in base al percorso utente

Questo esempio mostra come creare una policy basata sull'identità che consenta l'utilizzo della console del simulatore di policy solo per gli utenti con il percorso `Department/Development`. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

È possibile accedere al simulatore di policy IAM all'indirizzo https://policysim.aws.amazon.com/

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetPolicy",
                "iam:GetUserPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:GetUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListGroupsForUser",
                "iam:ListUserPolicies",
                "iam:ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:user/Department/Development/*"
        }
    ]
}
```

IAM: consente agli utenti IAM di gestire in modo autonomo un dispositivo MFA

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di gestire in modo automatico il proprio dispositivo di autenticazione a più fattori (MFA). Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

> **Note**
>
> Se un utente IAM con questa policy non è autenticato tramite MFA, questa policy nega l'accesso a tutte le AWS azioni tranne quelle necessarie per l'autenticazione tramite MFA. Se aggiungi queste autorizzazioni per un utente che ha effettuato l'accesso AWS, potrebbe dover disconnettersi e riconnettersi per visualizzare le modifiche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToCreateVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam:CreateVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/*"
        },
        {
            "Sid": "AllowUserToManageTheirOwnMFA",
            "Effect": "Allow",
            "Action": [
                "iam:EnableMFADevice",
                "iam:GetMFADevice",
                "iam:ListMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        },
        {
            "Sid": "AllowUserToDeactivateTheirOwnMFAOnlyWhenUsingMFA",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ],
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
```

```
                }
            }
        },
        {
            "Sid": "BlockMostAccessUnlessSignedInWithMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam:CreateVirtualMFADevice",
                "iam:EnableMFADevice",
                "iam:ListMFADevices",
                "iam:ListUsers",
                "iam:ListVirtualMFADevices",
                "iam:ResyncMFADevice"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": "false"
                }
            }
        }
    ]
}
```

IAM: consente agli utenti IAM di aggiornare le credenziali a livello di programmazione e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di aggiornare le proprie chiavi di accesso, i certificati di firma, le credenziali specifiche del servizio e le password. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        }
    ]
}
```

Per ulteriori informazioni su come un utente può modificare la propria password nella console, consultare the section called “Come un utente IAM può modificare la propria password”.

IAM: visualizzazione delle informazioni dell'ultimo accesso al servizio per una policy di Organizations

Questo esempio mostra come creare una policy basata sull'identità che consenta di visualizzare le informazioni sull'ultimo accesso per una determinata policy di Organizations. Questa policy consente di recuperare dati per la policy di controllo del servizio (SCP) con l'ID `p-policy123`. La persona che genera e visualizza il report deve essere autenticata utilizzando le credenziali dell'account di AWS Organizations gestione. Questa policy consente al richiedente di recuperare i dati per qualsiasi entità di Organizations nella propria organizzazione. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per informazioni importanti sui dati sull'ultimo accesso al servizio, incluse le autorizzazioni richieste, la risoluzione dei problemi e le regioni supportate, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOrgsReadOnlyAndIamGetReport",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:GetOrganizationsAccessReport",
                "organizations:Describe*",
                "organizations:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGenerateReportOnlyForThePolicy",
            "Effect": "Allow",
            "Action": "iam:GenerateOrganizationsAccessReport",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"iam:OrganizationsPolicyId": "p-policy123"}
            }
        }
    ]
}
```

IAM: limita le policy gestite che possono essere applicate a un utente, un gruppo o un ruolo IAM.

Questo esempio mostra come è possibile creare una policy basata sull'identità che limiti le policy gestite dai clienti e quelle AWS gestite che possono essere applicate a un utente, gruppo o ruolo IAM. Questa policy concede le autorizzazioni necessarie per completare questa azione in modo programmatico dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:AttachUserPolicy",
            "iam:DetachUserPolicy"
        ],
        "Resource": "*",
        "Condition": {
            "ArnEquals": {
                "iam:PolicyARN": [
                    "arn:aws:iam::*:policy/policy-name-1",
```

```
                    "arn:aws:iam::*:policy/policy-name-2"
                ]
            }
        }
    }
}
```

AWS: nega l'accesso alle risorse esterne al tuo account ad eccezione delle politiche IAM AWS gestite

L'utilizzo di `aws:ResourceAccount` nelle policy basate sull'identità può influire sulla capacità dell'utente o del ruolo di utilizzare alcuni servizi che richiedono l'interazione con le risorse negli account di proprietà di un servizio.

Puoi creare una policy con un'eccezione per consentire le policy IAM AWS gestite. Un account gestito dal servizio esterno alla tua AWS Organizations possiede le Managed IAM Policies. Esistono quattro azioni IAM che elencano e AWS recuperano le politiche gestite. Utilizza queste operazioni nell'elemento `NotAction` dell'istruzione `AllowAccessToS3ResourcesInSpecificAccountsAndSpecificService1` nella policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToResourcesInSpecificAccountsAndSpecificService1",
      "Effect": "Deny",
      "NotAction": [
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:ListEntitiesForPolicy",
        "iam:ListPolicies"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "111122223333"
          ]
        }
      }
    }
  ]
```

```
}
```

AWS Lambda: consente a una funzione Lambda di accedere a una tabella Amazon DynamoDB

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso in lettura e scrittura a una tabella Amazon DynamoDB specifica. La politica consente anche di scrivere file di registro in CloudWatch Logs. Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Per utilizzare questa policy, collega la policy a un ruolo del servizio Lambda. Un ruolo del servizio è un ruolo che viene creato nell'account per consentire a un servizio di eseguire operazioni a tuo nome. Tale ruolo di servizio deve essere incluso AWS Lambda come principale nella politica di fiducia. Per informazioni dettagliate su come utilizzare questa policy, consulta Come creare una policy AWS IAM per concedere AWS Lambda l'accesso a una tabella Amazon DynamoDB nel AWS Security Blog.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteTable",
            "Effect": "Allow",
            "Action": [
                "dynamodb:BatchGetItem",
                "dynamodb:GetItem",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem"
            ],
            "Resource": "arn:aws:dynamodb:*:*:table/SampleTable"
        },
        {
            "Sid": "GetStreamRecords",
            "Effect": "Allow",
            "Action": "dynamodb:GetRecords",
            "Resource": "arn:aws:dynamodb:*:*:table/SampleTable/stream/* "
        },
        {
```

```
            "Sid": "WriteLogStreamsAndGroups",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateLogGroup",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "*"
        }
    ]
}
```

Amazon RDS: consente l'accesso completo al database RDS in una regione specifica

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso completo al database RDS in una regione specifica. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rds:*",
            "Resource": ["arn:aws:rds:region:*:*"]
        },
        {
            "Effect": "Allow",
            "Action": ["rds:Describe*"],
            "Resource": ["*"]
        }
    ]
}
```

Amazon RDS: consente il ripristino dei database RDS, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta di ripristinare i database RDS. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSnapshot",
                "rds:DeleteDBSnapshot",
                "rds:Describe*",
                "rds:DownloadDBLogFilePortion",
                "rds:List*",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyOptionGroup",
                "rds:RebootDBInstance",
                "rds:RestoreDBInstanceFromDBSnapshot",
                "rds:RestoreDBInstanceToPointInTime"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon RDS: consente ai proprietari di tag l'accesso completo alle risorse RDS da loro contrassegnate con un tag

Questo esempio mostra come creare una policy basata sull'identità che conceda ai proprietari dei tag l'accesso completo alle risorse RDS che hanno contrassegnato con tag. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "rds:Describe*",
                "rds:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "rds:DeleteDBInstance",
                "rds:RebootDBInstance",
                "rds:ModifyDBInstance"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:db-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyOptionGroup",
                "rds:DeleteOptionGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:og-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBParameterGroup",
                "rds:ResetDBParameterGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:pg-tag/Owner": "${aws:username}"}
            }
        },
        {
```

```
            "Action": [
                "rds:AuthorizeDBSecurityGroupIngress",
                "rds:RevokeDBSecurityGroupIngress",
                "rds:DeleteDBSecurityGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:secgrp-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:DeleteDBSnapshot",
                "rds:RestoreDBInstanceFromDBSnapshot"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:snapshot-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBSubnetGroup",
                "rds:DeleteDBSubnetGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:subgrp-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyEventSubscription",
                "rds:AddSourceIdentifierToSubscription",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:DeleteEventSubscription"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:es-tag/Owner": "${aws:username}"}
```

```
            }
        }
    ]
}
```

Amazon S3: consente agli utenti di Amazon Cognito di accedere a oggetti nel relativo bucket

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti Amazon Cognito di accedere a oggetti in un determinato bucket S3. Questa policy consente l'accesso solo agli oggetti con un nome che include `cognito`, il nome dell'applicazione e l'ID dell'utente federato, rappresentati dalla variabile ${cognito-identity.amazonaws.com:sub}. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

> **Note**
>
> Il valore 'sub' utilizzato nella chiave dell'oggetto non è il valore secondario dell'utente nel pool di utenti, è l'ID identità associato all'utente nel pool di identità.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListYourObjects",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "cognito/application-name/${cognito-identity.amazonaws.com:sub}/*"
          ]
        }
      }
```

```
    },
    {
      "Sid": "ReadWriteDeleteYourObjects",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/cognito/application-name/${cognito-
identity.amazonaws.com:sub}/*"
      ]
    }
  ]
}
```

Amazon Cognito fornisce autenticazione, autorizzazione e gestione degli utenti per le app Web e per dispositivi mobili. Gli utenti possono accedere direttamente con un nome utente e una password, oppure tramite terze parti , ad esempio Facebook, Amazon o Google.

I due componenti principali di Amazon Cognito sono i bacini d'utenza e i pool di identità. I bacini d'utenza sono directory utente che forniscono opzioni di registrazione e di accesso agli utenti delle tue app. I pool di identità consentono di concedere agli utenti l'accesso ad altri servizi. AWS È possibile usare i pool di identità e i bacini d'utenza separatamente o insieme.

Per ulteriori informazioni su Amazon Cognito, consulta la Guida per l'utente di Amazon Cognito.

Amazon S3: consente agli utenti federati di accedere alla propria directory home S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti federati di accedere all'oggetto bucket nella loro directory home in S3. La directory iniziale è un bucket che include una cartella `home` e le cartelle per i singoli utenti federati. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

La variabile `${aws:userid}` in questa policy restituisce `role-id:specified-name`. La parte `role-id` dell'ID utente federato è un identificatore univoco assegnato al ruolo dell'utente federato durante la creazione. Per ulteriori informazioni, consulta Identificatori univoci. `specified-name`è

il RoleSessionName parametro passato alla `AssumeRoleWithWebIdentity` richiesta quando l'utente federato ha assunto il proprio ruolo.

È possibile visualizzare l'ID del ruolo utilizzando il AWS CLI comando`aws iam get-role --role-name` *`specified-name`*. Ad esempio, supponiamo di specificare il nome descrittivo `John` e che la CLI restituisca l'ID ruolo `AROAXXT2NJT7D3SIQN7Z6`. In questo caso, l'ID utente federato è `AROAXXT2NJT7D3SIQN7Z6:John`. Questa policy quindi consente all'utente federato John di accedere ai bucket Amazon S3 con il prefisso `AROAXXT2NJT7D3SIQN7Z6:John`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::bucket-name",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "",
                        "home/",
                        "home/${aws:userid}/*"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::bucket-name/home/${aws:userid}",
                "arn:aws:s3:::bucket-name/home/${aws:userid}/*"
            ]
        }
    ]
}
```

Amazon S3: accesso al bucket S3, ma bucket di produzione rifiutato senza MFA recente

Questo esempio mostra come creare una policy basata sull'identità che consenta a un amministratore Amazon S3 di accedere a qualsiasi bucket, inclusi l'aggiornamento, l'aggiunta e l'eliminazione di oggetti. Tuttavia, rifiuta esplicitamente l'accesso al bucket `Production` se l'utente non ha effettuato l'accesso utilizzando l'autenticazione multi-fattore (MFA) negli ultimi 30 minuti. Questo criterio concede le autorizzazioni necessarie per eseguire questa azione nella console o a livello di codice utilizzando l'API o. AWS CLI AWS Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy non consente mai l'accesso a livello di programmazione al bucket `Production` utilizzando le chiavi di accesso degli utenti a lungo termine. Questa operazione viene eseguita utilizzando la chiave di condizione `aws:MultiFactorAuthAge` con l'operatore di condizione `NumericGreaterThanIfExists`. Questa condizione di policy restituisce `true` se MFA non è presente o se l'età di MFA è superiore a 30 minuti. In tali situazioni, l'accesso è negato. Per accedere al `Production` bucket a livello di codice, l'amministratore S3 deve utilizzare credenziali temporanee generate negli ultimi 30 minuti utilizzando l'operazione API. GetSessionToken

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListAllS3Buckets",
            "Effect": "Allow",
            "Action": ["s3:ListAllMyBuckets"],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowBucketLevelActions",
```

```
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowBucketObjectActions",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::*/*"
        },
        {
            "Sid": "RequireMFAForProductionBucket",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::Production/*",
                "arn:aws:s3:::Production"
            ],
            "Condition": {
                "NumericGreaterThanIfExists": {"aws:MultiFactorAuthAge": "1800"}
            }
        }
    ]
}
```

Amazon S3: consente agli utenti IAM di accedere alla propria directory home S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti IAM di accedere al proprio oggetto del bucket nella directory home in S3. La home directory è un bucket che include una cartella home e le cartelle per i singoli utenti. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il

testo segnaposto in corsivo nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Questa policy non funziona quando si utilizzano i ruoli IAM perché la variabile `aws:username` non è disponibile quando si utilizzano i ruoli IAM. Per informazioni dettagliate sui valori delle chiavi principali, consulta Valori della chiave dell'entità principale.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::bucket-name",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "",
                        "home/",
                        "home/${aws:username}/*"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
```

```
                "arn:aws:s3:::bucket-name/home/${aws:username}",
                "arn:aws:s3:::bucket-name/home/${aws:username}/*"
            ]
        }
    ]
}
```

Amazon S3: limitazione della gestione a un bucket S3 specifico

Questo esempio mostra come creare una policy basata sull'identità che limiti la gestione di un bucket Amazon S3 a quel determinato bucket. Questa policy concede l'autorizzazione a eseguire tutte le operazioni di Amazon S3, ma nega l'accesso a ogni Servizio AWS tranne Amazon S3. Guarda l'esempio seguente. In base a questa policy, puoi accedere solo alle operazioni di Amazon S3 che è possibile eseguire su un bucket S3 o una risorsa oggetto S3. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *testo segnaposto in corsivo* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

Se questa politica viene utilizzata in combinazione con altre politiche (come le politiche FullAccess AWS gestite da AmazonS3 FullAccess o AmazonEC2) che consentono azioni negate da questa politica, l'accesso viene negato. Questo perché una dichiarazione di rifiuto esplicita prevale su una dichiarazione di consenso. Per ulteriori informazioni, consulta the section called “Determinazione se una richiesta è consentita o rifiutata in un account”.

> **Warning**
>
> `NotAction` e `NotResource` sono elementi di policy avanzate da utilizzare con attenzione. Questa policy rifiuta l'accesso a qualsiasi servizio AWS a eccezione di Amazon S3. Se colleghi questa policy a un utente, qualsiasi altra policy che concede le autorizzazioni ad altri servizi viene ignorata e l'accesso viene negato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
```

```
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
            "Effect": "Deny",
            "NotAction": "s3:*",
            "NotResource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

Amazon S3: consente l'accesso in lettura e scrittura agli oggetti di un bucket S3

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso `Read` e `Write` agli oggetti di un bucket S3 specifico. Questa politica concede le autorizzazioni necessarie per completare questa azione a livello di codice dall'API o. AWS AWS CLI Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'operazione `s3:*Object` usa un carattere jolly nel nome dell'operazione. L'istruzione `AllObjectActions` consente le operazioni `GetObject`, `DeleteObject`, `PutObject` e qualsiasi altra operazione Amazon S3 che termina con la parola "Object".

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::bucket-name"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        }
```

```
    ]
}
```

> **Note**
>
> Per consentire l'accesso `Read` e `Write` a un oggetto di un bucket Amazon S3 e includere anche altre autorizzazioni per l'accesso alla console, consulta Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console.

Amazon S3: consente l'accesso in lettura e scrittura agli oggetti in un bucket S3, in modo programmatico e nella console

Questo esempio mostra come creare una policy basata sull'identità che consenta l'accesso `Read` e `Write` agli oggetti di un bucket S3 specifico. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

L'operazione `s3:*Object` usa un carattere jolly nel nome dell'operazione. L'istruzione `AllObjectActions` consente le operazioni `GetObject`, `DeleteObject`, `PutObject` e qualsiasi altra operazione Amazon S3 che termina con la parola "Object".

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": ["arn:aws:s3:::bucket-name"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        }
    ]
}
```

Gestione di policy IAM

IAM fornisce gli strumenti per creare e gestire tutti i tipi di policy IAM (policy gestite e policy in linea). Per aggiungere le autorizzazioni a un'identità IAM (utente, gruppo o ruolo IAM), crea una policy, convalida la policy, quindi collega la policy all'identità. È possibile anche collegare più policy a un'entità e ogni policy può contenere più autorizzazioni.

Consulta queste risorse per i dettagli:

- Per ulteriori informazioni sui diversi tipi di policy IAM, consulta Policy e autorizzazioni in IAM.
- Per informazioni generali sull'uso delle policy con IAM, consulta Gestione degli accessi AWS alle risorse.
- Per ulteriori informazioni su come vengono valutate le autorizzazioni quando vengono applicate più policy per una determinata identità IAM, consulta Logica di valutazione delle policy.
- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Argomenti

- Creazione di policy IAM
- Convalida delle policy IAM
- Generazione di policy basate sull'attività di accesso
- Test delle policy IAM con il simulatore di policy IAM

- Aggiunta e rimozione di autorizzazioni per identità IAM
- Controllo delle versioni delle policy IAM
- Modifica delle policy IAM
- Eliminazione di policy IAM
- Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso

Creazione di policy IAM

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare l' AWS API AWS Management Console AWS CLI, o per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Puoi quindi collegare le politiche alle identità (utenti, gruppi e ruoli) nel tuo Account AWS.

Una policy collegata a un'identità in IAM è nota come policy basata sull'identità. Le politiche basate sull'identità possono includere politiche gestite, politiche AWS gestite dai clienti e politiche in linea. AWS le politiche gestite vengono create e gestite da. AWS Puoi usarle, ma non puoi gestirle. Una policy in linea è una policy che viene creata e integrata direttamente in un gruppo, utente o ruolo IAM. Le policy inline non possono essere riutilizzate su altre identità o gestite al di fuori dell'identità in cui esistono. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Utilizzo delle policy gestite dal cliente invece delle policy inline. È inoltre preferibile utilizzare politiche gestite dal cliente anziché politiche AWS gestite. AWS le politiche gestite in genere forniscono ampie autorizzazioni amministrative o di sola lettura. Per garantire la massima sicurezza, concedere un privilegio minimo, ovvero concedere solo le autorizzazioni necessarie per eseguire attività di lavoro specifiche.

Quando crei o modifichi le policy IAM, AWS puoi eseguire automaticamente la convalida delle policy per aiutarti a creare una policy efficace con il minimo privilegio in mente. Nel AWS Management Console, IAM identifica gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce controlli aggiuntivi sulle policy con consigli per aiutarti a perfezionare ulteriormente le tue policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Puoi utilizzare l' AWS API AWS Management Console AWS CLI, o per creare policy gestite dai clienti in IAM. Per ulteriori informazioni sull'utilizzo dei AWS CloudFormation modelli per aggiungere o

aggiornare le politiche, consulta il riferimento ai tipi di AWS Identity and Access Management risorse nella Guida per l'AWS CloudFormation utente.

Argomenti

- Creazione di policy IAM (console)
- Creazione di policy IAM (AWS CLI)
- Creazione di politiche IAM (AWS API)

Creazione di policy IAM (console)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzarli AWS Management Console per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Puoi quindi allegare le politiche alle identità (utenti, gruppi e ruoli) nel tuo Account AWS.

Argomenti

- Creazione di policy IAM
- Creazione di policy utilizzando l'editor JSON
- Creazione di policy con l'editor visivo
- L'importazione di policy gestite esistenti

Creazione di policy IAM

È possibile creare una politica gestita dai clienti AWS Management Console utilizzando uno dei seguenti metodi:

- JSON: incolla e personalizza un esempio di policy basata sull'identità.
- Editor visivo: è possibile creare una nuova policy da zero nell'editor visivo. Se si utilizza l'editor visivo, non è necessario comprendere la sintassi JSON.
- Importa: importa e personalizza una policy gestita dall'account. È possibile importare una politica AWS gestita o una politica gestita dai clienti creata in precedenza.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di policy utilizzando l'editor JSON

Puoi digitare o incollare le policy in JSON scegliendo l'opzione JSON. Questo metodo è utile per copiare una policy di esempio da utilizzare nell'account. In alternativa, è possibile digitare il proprio documento di policy JSON nell'editor JSON. È inoltre possibile utilizzare l'opzione JSON per passare tra l'editor visivo e JSON e confrontare le visualizzazioni.

Quando si crea o si modifica una policy nell'editor JSON, IAM esegue la convalida delle policy per facilitare la creazione di una policy efficace. IAM identifica gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti utili per perfezionare ulteriormente la policy.

Un documento di policy JSON consiste in una o più istruzioni. Ogni istruzione deve contenere tutte le operazioni che condividono lo stesso risultato (`Allow` o `Deny`) e supportare le stesse risorse e condizioni. Se un'operazione richiede di specificare tutte le risorse (`"*"`) e un'altra operazione supporta l'Amazon Resource Name (ARN) di una risorsa specifica, devono essere in due diverse istruzioni JSON. Per informazioni dettagliate sui formati ARN, consulta Amazon Resource Name (ARN) nella Guida Riferimenti generali di AWS . Per informazioni generali sulle policy IAM, consulta Policy e autorizzazioni in IAM. Per informazioni sul linguaggio delle policy IAM, consulta Riferimento alla policy JSON IAM.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Digitare o incollare un documento di policy JSON. Per maggiori dettagli sul linguaggio della policy IAM, consulta Riferimento alla policy JSON IAM.
6. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe

ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

7. (Facoltativo) Quando crei o modifichi una policy in AWS Management Console, puoi generare un modello di policy JSON o YAML da utilizzare nei modelli. AWS CloudFormation

   Per fare ciò, nell'editor delle politiche scegli Azioni, quindi scegli Genera modello. CloudFormation Per ulteriori informazioni, AWS CloudFormation consulta il riferimento al tipo di AWS Identity and Access Management risorsa nella Guida AWS CloudFormation per l'utente.
8. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
9. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
10. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
11. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Creazione di policy con l'editor visivo

L'editor visivo nella console IAM fornisce informazioni utili sulla creazione di una policy senza dover scrivere una sintassi JSON. Per visualizzare un esempio dell'editor visivo per creare una policy, consultare the section called “Controllo dell'accesso alle identità”.

Per utilizzare l'editor visivo per creare una policy.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Policy editor, trova la sezione Seleziona un servizio, quindi scegli un AWS servizio. È possibile utilizzare la casella di ricerca in alto per limitare i risultati nell'elenco di servizi. È possibile selezionare solo un servizio nel blocco di autorizzazione di un editor visivo. Per concedere l'accesso a più di un servizio, aggiungi più blocchi di autorizzazioni selezionando Aggiungi altre autorizzazioni.

5. In Operazioni consentite, scegli le operazioni da aggiungere alla policy. È possibile selezionare operazioni nei modi seguenti:

   - Selezionare la casella di controllo per tutte le azioni.
   - Scegliere aggiungi azioni per digitare il nome di un'azione specifica. È possibile utilizzare i caratteri jolly (*) per specificare più operazioni.
   - Selezionare uno dei gruppi di livelli di accesso per scegliere tutte le azioni per il livello di accesso, ad esempio Lettura, Scrittura o Elenco.
   - Espandere ciascuno dei gruppi Access level (Livello di accesso) per selezionare singole operazioni.

   Come impostazione predefinita, la policy che si sta creando utilizza le operazioni selezionate. Per rifiutare invece le operazioni scelte, selezionare Switch to deny permissions (Passa a rifiuto autorizzazioni). Poiché IAM rifiuta per impostazione predefinita, si consiglia come best practice di sicurezza di consentire le autorizzazioni solo alle operazioni e alle risorse necessarie per un utente. È necessario creare un'istruzione JSON per negare le autorizzazioni solo se si desidera sostituire un'autorizzazione separatamente consentita da un'altra istruzione o policy. Si consiglia di limitare al minimo il numero di autorizzazioni di rifiuto perché possono aumentare la difficoltà di risoluzione dei problemi relative alle autorizzazioni.

6. Per Risorse, se il servizio e le azioni selezionati nei passaggi precedenti non supportano la scelta di risorse specifiche, tutte le risorse sono consentite e non è possibile modificare questa sezione.

   Se si selezionano una o più operazioni che supportano le autorizzazioni a livello di risorsa, l'editor visivo elenca tali risorse. È possibile selezionare Risorse per specificare le risorse per la policy.

   È possibile specificare le risorse nei seguenti modi:

   - Seleziona Aggiungi ARN per specificare le risorse in base al loro nome della risorsa Amazon (ARN). È possibile utilizzare l'editor ARN visivo o elencare manualmente gli ARN. Per maggiori informazioni sulla sintassi ARN, consulta Amazon Resource Name (ARN) nella Guida Riferimenti generali di AWS . Per informazioni sull'utilizzo di ARN nell'elemento `Resource` di una policy, consulta Elementi delle policy JSON IAM: Resource.
   - Scegli Qualsiasi in questo account accanto a una risorsa per concedere autorizzazioni a qualsiasi risorsa di quel tipo.
   - Seleziona Tutto per selezionare tutte le risorse per quel servizio.

7. (Facoltativo) Scegli Condizioni di richiesta - opzionale per aggiungere condizioni alla policy che si sta creando. Le condizioni limitano l'effetto di una dichiarazione di policy JSON. Ad esempio, puoi specificare che un utente può eseguire le operazioni sulle risorse solo quando la richiesta dell'utente viene effettuata entro un determinato intervallo di tempo. È inoltre possibile utilizzare le condizioni comuni per limitare se un utente deve essere autenticato utilizzando un dispositivo multi-factor authentication (MFA). In alternativa, è possibile richiedere che la richiesta provenga da un determinato intervallo di indirizzi IP. Per un elenco di tutte le chiavi di contesto che è possibile utilizzare in una condizione di policy, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

   È possibile selezionare le condizioni nei modi seguenti:

   - Utilizzare le caselle di controllo per selezionare le condizioni di utilizzo comune.
   - Seleziona Aggiungi altra condizione per specificare altre condizioni. Selezionare Condition Key (Chiave condizione), Qualifier (Qualificatore) e Operator (Operatore) della condizione e digitare un Value (Valore). Per aggiungere più di un valore, seleziona Aggiungi. È possibile valutare i valori come se fossero connessi da un operatore logico "OR". Una volta terminato, scegli Aggiungi condizione.

   Per aggiungere più di una condizione, scegli di nuovo Aggiungi altra condizione. Ripetere come necessario. Ogni condizione si applica solo a questo blocco di autorizzazione di un editor visivo. Tutte le condizioni devono essere vere per il blocco di autorizzazioni per essere considerato una corrispondenza. In altre parole, considerare le condizioni da connettere con un operatore logico "AND".

   Per ulteriori informazioni sull'elemento Condition (Condizione), consultare Elementi delle policy JSON IAM: Condition in Riferimento alla policy JSON IAM.

8. Per aggiungere più blocchi di autorizzazioni, seleziona Aggiungi ulteriori autorizzazioni. Per ogni blocco, ripetere le fasi da 2 a 5.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

9. (Facoltativo) Quando si crea o si modifica una politica in AWS Management Console, è possibile generare un modello di policy JSON o YAML da utilizzare nei modelli. AWS CloudFormation

   Per fare ciò, nell'editor delle politiche scegli Azioni, quindi scegli Genera modello. CloudFormation Per ulteriori informazioni, AWS CloudFormation consulta il riferimento al tipo di AWS Identity and Access Management risorsa nella Guida AWS CloudFormation per l'utente.
10. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
11. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Rivedi il campo Autorizzazioni definite in questa policy per accertarti di disporre delle autorizzazioni previste.
12. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consultare Tagging delle risorse IAM.
13. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

L'importazione di policy gestite esistenti

Un modo semplice per creare una nuova policy è di importare una policy gestita esistente all'interno dell'account che dispone di almeno alcune delle autorizzazioni di cui si ha bisogno. È possibile personalizzare la policy per farla corrispondere ai nuovi requisiti.

Non è possibile importare una policy inline. Per informazioni sulle differenze tra policy gestite e policy inline, consultare Policy gestite e policy inline.

Per importare una policy gestita esistente nell'editor visivo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione Visivo, quindi sul lato destro della pagina, scegli Operazioni e poi Importa policy.
5. Nella finestra Importa policy gestite, seleziona le policy gestite che meglio corrispondono alla policy da includere nella nuova policy. Per limitare i risultati nell'elenco di servizi, è possibile utilizzare la casella di ricerca in alto.

6. Scegli Importa policy.

   Le policy importate vengono aggiunte in nuovi blocchi di autorizzazione nella parte inferiore della policy.

7. Utilizzare Visual editor (Editor visivo) o selezionare JSON per personalizzare la policy. Quindi scegli Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

8. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Non è possibile modificare queste impostazioni in un secondo momento. Rivedi il campo Autorizzazioni definite in questa policy, quindi scegli Crea policy per salvare il lavoro.

Importazione di una policy gestita esistente nell'editor JSON

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON, quindi sul lato destro della pagina, scegli Operazioni e poi Importa policy.
5. Nella finestra Importa policy gestite, seleziona le policy gestite che meglio corrispondono alla policy da includere nella nuova policy. Per limitare i risultati nell'elenco di servizi, è possibile utilizzare la casella di ricerca in alto.
6. Scegli Importa policy.

   Le istruzioni dalle policy importate vengono aggiunte in fondo alle policy JSON.

7. Personalizza la policy in JSON. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Oppure, personalizza la policy nell'Editor visivo. Quindi scegli Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

8. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Non è possibile modificare queste impostazioni in un secondo momento. Rivedi la policy Autorizzazioni definite in questa policy, quindi scegli Crea policy per salvare il lavoro.

Dopo aver creato una policy, è possibile collegarlo ai gruppi, utenti o ruoli. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Creazione di policy IAM (AWS CLI)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzarli AWS CLI per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Attraverso la convalida delle policy è possibile risolvere eventuali errori o suggerimenti prima di collegare le policy alle identità (utenti, gruppi e ruoli) dell' Account AWS.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di policy IAM (AWS CLI)

Puoi creare una policy gestita dal cliente IAM o una policy in line utilizzando AWS Command Line Interface (AWS CLI).

Per creare una policy gestita dal cliente (AWS CLI)

Utilizza il seguente comando:

- create-policy

Come creare una policy in linea per un'identità IAM (utente, gruppo o ruolo) (AWS CLI)

Utilizzare uno dei seguenti comandi:

- put-group-policy
- put-role-policy
- put-user-policy

> **Note**
>
> Non è possibile utilizzare IAM per integrare una policy in linea per un ruolo collegato ai servizi.

Come convalidare una policy gestita dal cliente (AWS CLI)

Utilizza il seguente comando IAM Access Analyzer:

- validate-policy

Creazione di politiche IAM (AWS API)

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Puoi utilizzare l' AWS API per creare policy gestite dai clienti in IAM. Le policy gestite dal cliente sono policy autonome gestite dall'utente nel proprio Account AWS. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Attraverso la convalida delle policy è possibile risolvere eventuali errori o suggerimenti prima di collegare le policy alle identità (utenti, gruppi e ruoli) dell' Account AWS.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Creazione di politiche IAM (AWS API)

Puoi creare una policy gestita dal cliente IAM o una policy in linea utilizzando l'API AWS .

Per creare una policy gestita dai clienti (AWS API)

Chiamare l'operazione seguente:

- CreatePolicy

Come creare una policy in linea per un'identità IAM (utente, gruppo o ruolo) (API AWS)

Chiamare una delle seguenti operazioni:

- PutGroupPolicy
- PutRolePolicy
- PutUserPolicy

> **Note**
>
> Non è possibile utilizzare IAM per integrare una policy in linea per un ruolo collegato ai servizi.

Per convalidare una politica gestita dal cliente (AWS API)

Chiama la seguente operazione IAM Access Analyzer:

- ValidatePolicy

Convalida delle policy IAM

Una policy è un documento JSON che utilizza la sintassi delle policy IAM. Quando colleghi una policy a un'entità IAM, come un utente, un gruppo o un ruolo, la policy concede le autorizzazioni a tale entità.

Quando crei o modifichi le policy di controllo degli accessi IAM utilizzando AWS Management Console, le esamina AWS automaticamente per garantire che siano conformi alla grammatica delle policy IAM. Se AWS determina che una policy non rispetta la sintassi, verrà richiesto di correggere la policy.

IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente la policy. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer. Per visualizzare un elenco di avvisi, errori e suggerimenti restituiti da IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

Ambito della convalida

AWS controlla la sintassi e la grammatica della policy JSON. Inoltre verifica che gli ARN siano formattati correttamente e che i nomi delle operazioni e le chiavi di condizione siano corretti.

Accesso alla convalida delle policy

Le policy vengono convalidate automaticamente quando si crea una policy JSON o si modifica una policy esistente nella AWS Management Console. Se la sintassi della policy non è valida, riceverai una notifica e dovrai correggere il problema prima di poter continuare. I risultati della convalida delle policy di IAM Access Analyzer vengono restituiti automaticamente nella cartella AWS Management Console if you have permissions for. `access-analyzer:ValidatePolicy` Puoi anche convalidare le politiche utilizzando l'API o. AWS AWS CLI

Policy esistenti

È possibile che le policy esistenti non siano valide perché sono state create o salvate per l'ultima volta prima degli ultimi aggiornamenti del motore di policy. Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Consigliamo di aprire le policy esistenti e rivedere i risultati della convalida della policy generati. Non è possibile modificare e salvare le policy esistenti senza correggere eventuali errori di sintassi della policy.

Generazione di policy basate sull'attività di accesso

In qualità di amministratore o sviluppatore, puoi concedere autorizzazioni a entità IAM (utenti o ruoli) che vanno oltre quanto richiesto. IAM fornisce diverse opzioni che consentono di perfezionare le autorizzazioni concesse. Un'opzione consiste nel generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy con autorizzazioni granulari che concedono solo le autorizzazioni necessarie per supportare il caso d'uso specifico.

Prendiamo l'esempio di uno sviluppatore il cui team di ingegneria ha lavorato a un progetto per creare una nuova applicazione. Per incoraggiare la sperimentazione e consentire al team di muoversi rapidamente, è stato configurato un ruolo con autorizzazioni generali mentre l'applicazione è in fase di sviluppo. Ora l'applicazione è pronta per la produzione. Prima che l'applicazione possa essere avviata nell'account di produzione, è necessario identificare solo le autorizzazioni necessarie al ruolo per il funzionamento dell'applicazione. In questo modo è più facile rispettare le best practice dei privilegi minimi. È possibile generare una policy basata sull'attività di accesso del ruolo utilizzato per

l'applicazione nell'account di sviluppo. È possibile perfezionare ulteriormente la policy generata e quindi allegare la policy nel proprio account di produzione.

Per ulteriori informazioni sulla generazione delle policy di IAM Access Analyzer, consulta Generazione di policy di IAM Access Analyzer.

Test delle policy IAM con il simulatore di policy IAM

Per ulteriori informazioni su come è perché utilizzare le policy IAM, consulta Policy e autorizzazioni in IAM.

È possibile accedere alla console del simulatore di policy di IAM all'indirizzo https://policysim.aws.amazon.com/

> **Important**
>
> I risultati del simulatore di policy possono differire dal tuo AWS ambiente reale. Ti consigliamo di verificare le tue politiche rispetto AWS all'ambiente reale dopo averle testate utilizzando il simulatore di policy per confermare di avere i risultati desiderati. Per ulteriori informazioni, consulta Come funziona il simulatore di policy IAM.

Nozioni di base sul simulatore di policy IAM

Con il simulatore di policy IAM è possibile testare e risolvere i problemi relativi a policy basate sulle identità e limiti delle autorizzazioni IAM. Di seguito sono elencate alcune operazioni comuni che è possibile eseguire con il simulatore di criteri:

- Esegui il test delle policy collegate a utenti, gruppi di utenti o ruoli IAM nell' Account AWS. Se più di una policy è collegata all'utente, al gruppo di utenti o al ruolo, è possibile testare tutte le policy oppure selezionare le policy individuali da testare. È possibile testare quali azioni sono consentite o negate dalle policy selezionate per le risorse specifiche.
- Verifica e risolvi i problemi relativi all'effetto dei limiti delle autorizzazioni sulle entità IAM. È possibile simulare solo un limite delle autorizzazioni alla volta.
- Esegui il test delle policy basate sulle risorse su utenti IAM collegati a AWS , ad esempio i bucket Amazon S3, le code Amazon SQS, gli argomenti di Amazon SNS o gli insiemi di credenziali di Amazon S3 Glacier. Per utilizzare una policy basata sulle risorse nel simulatore per gli utenti IAM, è necessario includere la risorsa nella simulazione. È inoltre necessario selezionare la casella di controllo per includere i criteri di tale risorsa nella simulazione.

> **Note**
>
> La simulazione di policy basate sulle risorse non è supportata per i ruoli IAM.

- Se fai Account AWS parte di un'organizzazione in AWS Organizations, puoi testare l'impatto delle policy di controllo dei servizi (SCP) sulle tue politiche basate sull'identità.

> **Note**
>
> Il simulatore di policy non valuta le policy di controllo dei servizi con condizioni.

- Esegui il test di nuove policy basate sull'identità che non sono ancora collegate a un utente, a un gruppo di utenti o a un ruolo digitandole o copiandole nel simulatore. Queste vengono utilizzate nella simulazione e non vengono salvate. Non è possibile digitare o copiare una policy basata su risorse nel simulatore.
- Esegui il test delle policy basate sull'identità con i servizi, le operazioni e le risorse selezionati. Ad esempio, puoi eseguire il test per assicurarti che la policy consenta a un'entità di eseguire le operazioni `ListAllMyBuckets`, `CreateBucket` e `DeleteBucket` nel servizio Amazon S3 su un determinato bucket.
- Simulare scenari reali fornendo chiavi di contesto, ad esempio un indirizzo IP o una data, inclusi in elementi `Condition` nella policy testate.

> **Note**
>
> Il simulatore di policy non simula i tag forniti come input se la policy basata sull'identità nella simulazione non ha un elemento `Condition` che controlli esplicitamente i tag.

- Identifica quali istruzioni specifiche in una policy risultano nel consenso o nella negazione dell'accesso a un'operazione o risorsa specifica.

Argomenti

Come funziona il simulatore di policy IAM

Il simulatore di policy valuta le dichiarazioni contenute nella policy basata sull'identità e gli input forniti durante la simulazione. I risultati del simulatore di policy possono differire dal tuo ambiente AWS reale. Ti consigliamo di verificare le tue politiche rispetto al tuo AWS ambiente reale dopo averle testate utilizzando il simulatore di policy per confermare di avere i risultati desiderati.

Il simulatore di policy si differenzia dall' AWS ambiente live nei seguenti modi:

- Il simulatore di policy non effettua una richiesta di AWS servizio effettiva, quindi puoi testare in sicurezza le richieste che potrebbero apportare modifiche indesiderate al tuo ambiente live. AWS Il simulatore di policy non considera i valori chiave del contesto reale nella produzione.
- Poiché il simulatore non simula l'esecuzione delle azioni selezionate, non può segnalare alcuna risposta alla richiesta simulata. L'unico risultato restituito è se l'operazione richiesta è consentita o negata.
- Se si modifica una policy nel simulatore, queste modifiche riguarderanno solo il simulatore. La politica corrispondente nella vostra azienda Account AWS rimane invariata.
- Non è possibile eseguire il test delle policy di controllo dei servizi con condizioni.
- Il simulatore di policy non supporta la simulazione per i ruoli IAM e gli utenti per l'accesso multi-account.

> **Note**
>
> Il simulatore di policy IAM non determina quali servizi supportano le chiavi di condizione globali per l'autorizzazione. Ad esempio, il simulatore di policy non identifica che un servizio non supporta `aws:TagKeys`.

Autorizzazioni necessarie per l'utilizzo del simulatore di policy IAM

È possibile utilizzare la console del simulatore di policy o l'API del simulatore di policy per testare le policy. Per impostazione predefinita, gli utenti della console possono testare le policy che non sono ancora collegate a un utente, a un gruppo di utenti o a un ruolo digitandole o copiandole nella console del simulatore di policy. Queste regole vengono utilizzate nella simulazione e non rivelano informazioni sensibili. Gli utenti API devono avere le autorizzazioni per testare le policy non associate. Puoi consentire agli utenti di console o API di testare le policy associate a utenti, gruppi di utenti o ruoli IAM nell' Account AWS. A tale scopo, è necessario fornire l'autorizzazione per

recuperare tali criteri. Per testare policy basate su risorse, gli utenti devono avere l'autorizzazione di recuperare la policy della risorsa.

Per esempi di policy di console o API che consentono a un utente di simulare le policy, consultare the section called “Politiche di esempio: AWS Identity and Access Management (IAM)”.

Autorizzazioni necessarie per l'utilizzo della console del simulatore di policy

Puoi consentire agli utenti di testare le policy collegate a utenti, gruppi o ruoli IAM nel tuo Account AWS. A tale scopo, è necessario fornire agli utenti le autorizzazioni per recuperare tali criteri. Per testare policy basate su risorse, gli utenti devono avere l'autorizzazione di recuperare la policy della risorsa.

Per visualizzare un esempio di policy che consente di utilizzare la console del simulatore di policy per policy associate a un utente, a un gruppo di utenti o a un ruolo, consulta IAM: accesso alla console del simulatore di policy.

Per visualizzare una policy di esempio che consente l'utilizzo della console del simulatore della policy solo agli utenti con un percorso specifico, consultare IAM: accesso alla console del simulatore di policy in base al percorso utente.

Per creare una policy per consentire l'utilizzo della console del simulatore di policy per solo un tipo di entità, utilizzare le seguenti procedure.

Per consentire agli utenti della console di simulare le policy per gli utenti

Includere le seguenti operazioni nella policy:

- `iam:GetGroupPolicy`
- `iam:GetPolicy`
- `iam:GetPolicyVersion`
- `iam:GetUser`
- `iam:GetUserPolicy`
- `iam:ListAttachedUserPolicies`
- `iam:ListGroupsForUser`
- `iam:ListGroupPolicies`
- `iam:ListUserPolicies`
- `iam:ListUsers`

Come consentire agli utenti della console di simulare le policy per i gruppi di utenti

Includere le seguenti operazioni nella policy:

- `iam:GetGroup`
- `iam:GetGroupPolicy`
- `iam:GetPolicy`
- `iam:GetPolicyVersion`
- `iam:ListAttachedGroupPolicies`
- `iam:ListGroupPolicies`
- `iam:ListGroups`

Per consentire agli utenti della console di simulare le policy per i ruoli

Includere le seguenti operazioni nella policy:

- `iam:GetPolicy`
- `iam:GetPolicyVersion`
- `iam:GetRole`
- `iam:GetRolePolicy`
- `iam:ListAttachedRolePolicies`
- `iam:ListRolePolicies`
- `iam:ListRoles`

Per testare policy basate su risorse, gli utenti devono avere l'autorizzazione di recuperare la policy della risorsa.

Come consentire agli utenti della console di testare le policy basate su risorse in un bucket Amazon S3

Includere la seguente operazione nella policy:

- `s3:GetBucketPolicy`

Ad esempio, la policy seguente utilizza questa operazione per consentire agli utenti della console di simulare una policy basata sulle risorse in un bucket Amazon S3 specifico.

```
{
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": "s3:GetBucketPolicy",
            "Resource":"arn:aws:s3:::bucket-name/*"
          }
        ]
      }
```

Autorizzazioni necessarie per l'utilizzo dell'API del simulatore di policy

L'API Policy Simulator GetContextKeyForCustomPolicyfunziona e SimulateCustomPolicyconsente di testare policy che non sono ancora associate a un utente, gruppo di utenti o ruolo. Per testare tali criteri, è possibile passare i criteri come stringhe all'API. Queste regole vengono utilizzate nella simulazione e non rivelano informazioni sensibili. Puoi inoltre utilizzare l'API per verificare le policy associate a utenti, gruppi di utenti o ruoli IAM nell' Account AWS. A tale scopo, è necessario fornire agli utenti le autorizzazioni per chiamare GetContextKeyForPrincipalPolicye. SimulatePrincipalPolicy

Per visualizzare un esempio di policy che consente di utilizzare l'API Policy Simulator per le policy allegate e non collegate nella versione corrente Account AWS, consulta. IAM: accesso all'API del simulatore di policy

Per creare una policy che consente l'utilizzo dell'API del simulatore di policy per solo un tipo di policy, utilizzare le seguenti procedure.

Per consentire agli utenti di un'API di simulare le policy passate direttamente all'API come stringhe

Includere le seguenti operazioni nella policy:

- `iam:GetContextKeysForCustomPolicy`
- `iam:SimulateCustomPolicy`

Come consentire agli utenti di un'API di simulare le policy collegate a utenti, gruppi, ruoli o risorse IAM

Includere le seguenti operazioni nella policy:

- `iam:GetContextKeysForPrincipalPolicy`

- `iam:SimulatePrincipalPolicy`

Ad esempio, per offrire a un utente di nome Bob l'autorizzazione a simulare una policy che è assegnata a un utente di nome Alice, fornire accesso a Bob alla seguente risorsa: `arn:aws:iam::777788889999:user/alice`.

Per visualizzare una policy di esempio che consente l'utilizzo dell'API del simulatore della policy solo agli utenti con un percorso specifico, consultare IAM: accesso all'API simulatore di policy basata sul percorso degli utenti.

Utilizzo del simulatore di policy IAM (console)

Per impostazione predefinita, gli utenti possono testare le policy che non sono ancora collegate a un utente, a un gruppo di utenti o a un ruolo digitandole o copiandole nella console del simulatore di policy. Queste regole vengono utilizzate nella simulazione e non rivelano informazioni sensibili.

Come eseguire il test di una policy non collegata a un utente, un gruppo di utenti o un ruolo (console)

1. Apri la console del simulatore di policy IAM all'indirizzo https://policysim.aws.amazon.com/.
2. Nel menu Mode: (Modalità:) nella parte superiore della pagina, selezionare New Policy (Nuova policy).
3. In Policy Sandbox (Sandbox policy), selezionare Create New Policy (Crea nuova policy).
4. Digita o copia una policy nel simulatore e utilizza il simulatore come descritto di seguito.

Dopo avere ottenuto l'autorizzazione per utilizzare la console del simulatore di policy IAM, è possibile utilizzare il simulatore per testare un utente IAM, un gruppo di utenti, un ruolo o una policy di risorsa.

Come eseguire il test di una policy collegata a un utente, un gruppo di utenti o un ruolo (console)

1. Apri la console del simulatore di policy IAM all'indirizzo https://policysim.aws.amazon.com/.

> **Note**
>
> Per accedere al simulatore di policy come utente IAM, utilizza l'URL di accesso univoco per accedere alla AWS Management Console. Visita quindi https://policysim.aws.amazon.com/. Per ulteriori informazioni sull'accesso come utente IAM, consulta In che modo gli utenti IAM accedono a AWS.

Il simulatore si apre nella modalità Existing Policies (Policy esistenti) ed elenca gli utenti IAM nell'account in Users, Groups, and Roles (Utenti, gruppi e ruoli).

2. Selezionare l'opzione opportuna per la propria attività:

Per testare questo:	Esegui questa operazione:
Una policy collegata a un utente	Selezionare Users (Utenti) nell'elenco Users, Groups, and Roles (Utenti, gruppi e ruoli). Selezionare l'utente.
Policy collegata a un gruppo di utenti	Selezionare Groups (Gruppi) nell'elenco Users, Groups, and Roles (Utenti, gruppi e ruoli). Quindi, scegli il gruppo di utenti.
Una policy collegata a un ruolo	Selezionare Roles (Ruoli) nell'elenco Users, Groups, and Roles (Utenti, gruppi e ruoli). Selezionare il ruolo.
Una policy collegata a una risorsa	Per informazioni, consulta Step 9.
Una policy personalizzata per un utente, un gruppo di utenti o un ruolo	Scegli Crea nuova policy. Nel riquadro nuovi criteri digitare o incollare un criterio e quindi scegliere Applica.

> **Suggerimento**
>
> Per testare una policy collegata al gruppo, puoi avviare il simulatore di policy IAM direttamente dalla console IAM: nel pannello di navigazione, scegli Gruppi. Selezionare il nome del gruppo sul quale si desidera testare una policy e selezionare la scheda Permissions (Autorizzazioni). Scegli Simula.
> Per testare una policy gestita dal cliente collegata a un utente: nel riquadro di navigazione, selezionare Users (Utenti). Selezionare il nome dell'utente sul quale si desidera testare la policy. Selezionare la scheda Permissions (Autorizzazioni) ed espandere la policy che si desidera testare. Più a destra, selezionare Simulate policy (Simula policy). Simulatore di policy IAM si apre in una nuova finestra e viene visualizzata la policy selezionata nel riquadro Policy.

3. (Facoltativo) Se l'account è membro di un'organizzazione in AWS Organizations, seleziona la casella di controllo accanto a SCP AWS Organizations per includere SCP nella valutazione simulata. Le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o un'unità organizzativa (UO). L'SCP limita le autorizzazioni per le entità negli account membri. Se una SCP blocca un servizio o un'operazione, nessuna entità in tale account può accedere a quel servizio né eseguire questa operazione. Ciò è valido anche se un amministratore esplicitamente concede le autorizzazioni a quell'operazione o servizio tramite una policy IAM o delle risorse.

   Se l'account non è un membro di un'organizzazione, la casella di controllo non viene visualizzata.

4. (Facoltativo) Puoi eseguire il test di una policy impostata come limite delle autorizzazioni per un'entità IAM (utente o ruolo), ma non per i gruppi di utenti. Se un criterio limite delle autorizzazioni è attualmente impostato per l'entità, verrà visualizzato nel riquadro Criteri . È possibile impostare solo un limite delle autorizzazioni per un'entità. Per testare un limite di autorizzazioni diverso, è possibile creare un limite di autorizzazioni personalizzato. A tale scopo, scegliere Crea nuovo criterio. Viene aperto un nuovo riquadro Criteri . Nel menu, scegliere Criteri di limite autorizzazioni IAM personalizzate. Immettere un nome per il nuovo criterio e digitare o copiare un criterio nello spazio sottostante. Scegliere Applica per salvare il criterio. Quindi, scegliere Indietro per tornare al riquadro Criteri originale. Seleziona quindi la casella di controllo accanto al limite delle autorizzazioni che desideri utilizzare per la simulazione.

5. (Facoltativo) Puoi eseguire il test solo di un sottoinsieme di policy collegate a un utente, un gruppo di utenti o un ruolo. A tale scopo, nel riquadro Criteri deselezionare la casella di controllo accanto a ciascun criterio che si desidera escludere.

6. In Policy Simulator (Simulatore di policy), selezionare Select service (Seleziona servizio) e scegliere il servizio da testare. Selezionare Select actions (Seleziona operazioni) e scegliere una o più operazioni da testare. Sebbene i menu mostrino le opzioni disponibili per un solo servizio alla volta, tutti i servizi e le operazioni selezionati vengono visualizzati in Action Settings and Results (Impostazioni e risultati operazione).

7. (Facoltativo) Se una delle policy che scegli in Step 2 e Step 5 include condizioni con le chiavi della condizione globale di AWS, devi fornire i valori per tali chiavi. È possibile eseguire questa operazione espandendo la sezione Global Settings (Impostazioni globali) e digitando i valori per i nomi chiave visualizzati.

> **Warning**
>
> Se si lascia il valore di una condizione chiave vuoto, tale chiave viene ignorata durante la simulazione. In alcuni casi, questo genera un errore e non è possibile eseguire la simulazione. In altri casi, la simulazione viene eseguita, ma i risultati possono non essere affidabili. In questi casi, la simulazione non corrisponde alle condizioni reali che includono un valore per la chiave o la variabile della condizione.

8. (Facoltativo) Ogni operazione selezionata viene mostrata nell'elenco Action Settings and Results (Impostazioni e risultati operazione) con Not simulated (Non simulata) nella colonna Permission (Autorizzazione) finché effettivamente la simulazione non viene eseguita. Prima di eseguire la simulazione, è possibile configurare ciascuna operazione con una risorsa. Per configurare le operazioni individuali per un determinato scenario, selezionare la freccia per espandere la riga dell'operazione. Se l'operazione supporta le autorizzazioni a livello di risorsa, è possibile digitare l'Amazon Resource Name (ARN) della risorsa specifica di cui si desidera testare l'accesso. Come impostazione predefinita, ogni risorsa è impostata su un carattere jolly (*). È anche possibile specificare un valore per qualsiasi chiave di contesto della condizione. Come indicato in precedenza, le chiavi con valori vuoti vengono ignorate, ciò può provocare errori di simulazione o risultati inaffidabili.

   a. Selezionare la freccia accanto al nome dell'operazione per espandere ogni riga e configurare qualsiasi informazioni aggiuntive necessarie per simulare accuratamente l'operazione nel proprio scenario. Se l'operazione richiede autorizzazioni a livello di risorsa, è possibile digitare l'Amazon Resource Name (ARN) della risorsa specifica alla quale si desidera simulare l'accesso. Come impostazione predefinita, ogni risorsa è impostata su un carattere jolly (*).

   b. Se l'operazione supporta autorizzazioni a livello di risorsa, ma non le richiede, è possibile selezionare Add Resource (Aggiungi risorsa) per selezionare il tipo di risorsa che si desidera aggiungere alla simulazione.

   c. Se nessuna delle policy selezionate include un elemento`Condition` che fa riferimento a una chiave contestuale per il servizio di questa operazione, tale nome della chiave è visualizzato sotto l'operazione. È possibile specificare il valore da utilizzare durante la simulazione di quell'operazione per la risorsa specificata.

   Operazioni che richiedono diversi gruppi di tipi di risorse

Alcune operazioni richiedono diversi tipi di risorse in circostanze diverse. Ogni gruppo di tipi di risorse è associato a uno scenario. Se uno di questi si applica alla propria simulazione, selezionarlo e il simulatore richiederà i tipi di risorse appropriate per tale scenario. L'elenco seguente mostra ciascuna delle opzioni di scenari supportate e le risorse che è necessario definire per eseguire la simulazione.

Ognuno dei seguenti scenari di Amazon EC2 richiede che vengano specificate le risorse `instance`, `image` e `security-group`. Se il proprio scenario include un volume EBS, è necessario specificare quel `volume` come una risorsa. Se lo scenario di Amazon EC2 include un virtual private cloud (VPC), è necessario fornire la risorsa `network-interface`. Se include una sottorete IP, è necessario specificare la risorsa `subnet`. Per ulteriori informazioni sulle opzioni dello scenario di Amazon EC2, consulta Piattaforme supportatenella Guida per l'utente di Amazon EC2.

- EC2-VPC- InstanceStore

  istanza, immagine, gruppo di sicurezza, interfaccia di rete
- EC2-VPC- -Sottorete InstanceStore

  istanza, immagine, gruppo di sicurezza, interfaccia di rete, sottorete
- EC2-VPC-EBS

  istanza, immagine, gruppo di sicurezza, interfaccia di rete, volume
- EC2-VPC-EBS-Subnet

  istanza, immagine, gruppo di sicurezza, interfaccia di rete, sottorete, volume

9. (Facoltativo) Se si desidera includere una policy basate su risorse nella simulazione, è necessario selezionare le operazioni che si desidera simulare su quella risorsa Step 6. Espandere le righe per le operazioni selezionate e digitare il nome ARN della risorsa con una policy che si desidera simulare. Selezionare Include Resource Policy (Includi policy delle risorse) accanto alla casella di testo ARN. Il simulatore di policy IAM attualmente supporta le policy basate sulle risorse solo dai seguenti servizi: Amazon S3 (solo policy basate su risorse; le liste ACL non sono attualmente supportate), Amazon SQS, Amazon SNS e vault S3 Glacier sbloccati (i vault bloccati non sono attualmente supportati).
10. Selezionare Run Simulation (Esegui simulazione) nell'angolo in alto a destra.

La colonna Permission (Autorizzazione) in ogni riga di Action Settings e Results (Impostazioni e risultati operazione) visualizza il risultato di simulazione di tale operazione nella risorsa specificata.

11. Per consultare quale istruzione in una policy ha esplicitamente permesso o negato un'operazione, selezionare il collegamento ***N*** matching statement(s) (Istruzioni corrispondenti) nella colonna Permissions (Autorizzazioni) per espandere la riga. Selezionare il collegamento Show statement (Mostra istruzione). Il riquadro Policies (Policy) mostra la policy rilevante con l'istruzione che ha interessato i risultati della simulazione evidenziata.

> **Note**
>
> Se un'operazione è implicitamente rifiutata: ovvero, se l'operazione è rifiutata solo perché non è consentito in modo esplicito; le opzioni Elenco e Mostra istruzione non vengono visualizzate.

Risoluzione dei problemi dei messaggi della console del simulatore di policy IAM

La tabella seguente elenca i messaggi informativi e di avviso che possono essere restituiti quando si utilizza il simulatore delle policy IAM. La tabella fornisce inoltre una procedura per risolverli.

Messaggio	Procedura per la risoluzione
Questa policy è stata modificata. Le modifiche non saranno salvate al tuo account.	Nessuna operazione necessaria. Questo messaggio è informativo. Se modifichi una policy esistente nel simulatore di policy IAM, tale modifica non influisce sull' Account AWS. Il simulatore di policy consente di modificare le policy solo a scopo di test.
Impossibile ottenere le policy delle risorse. Motivo: *`messaggio di errore dettagliato`*	Il simulatore di policy non è in grado di accedere a una policy basata sulle risorse necessaria. Verificare che il nome ARN specificato della risorsa sia corretto e che l'utente che esegue la simulazione abbia

Messaggio	Procedura per la risoluzione
	l'autorizzazione di leggere la policy della risorsa.
Una o più policy richiedono valori nelle impostazioni della simulazione. La simulazione potrebbe non riuscire senza questi valori.	Questo messaggio viene visualizzato se la policy che si sta testando contiene variabili o chiavi di condizione, ma non sono stati forniti valori per queste chiavi o variabili in Simulation Settings (Impostazioni simulazione). Per chiudere questo messaggio, selezionare Impostazioni simulazione e digitare un valore per ogni variabile o chiave della condizione.
Sono state modificate delle policy. Questi risultati non sono più validi.	Questo messaggio viene visualizzato se si modifica la policy selezionata mentre i risultati sono visualizzati nel riquadro Results (Risultati). I risultati illustrati nel riquadro Results (Risultati) non sono aggiornati dinamicamente. Per chiudere questo messaggio, selezionare nuovamente Run Simulation (Esegui simulazione) per visualizzare i nuovi risultati di simulazione in base alle modifiche apportate nel riquadro Policies (Policy).

Messaggio	Procedura per la risoluzione
La risorsa digitata per questa simulazione non corrisponde a questo servizio.	Questo messaggio viene visualizzato se è stato digitato un Amazon Resource Name (ARN) nel riquadro Simulation Settings (Impostazioni simulazione) che non corrisponde al servizio scelto per la simulazione corrente. Ad esempio, questo messaggio viene visualizzato se viene specificato un ARN per una risorsa Amazon DynamoDB, ma si seleziona Amazon Redshift come servizio da simulare. Per chiudere questo messaggio, procedere in uno dei seguenti modi: • Rimuovere l'ARN dalla casella nel riquadro Simulation Settings (Impostazioni simulazio ne). • Selezionare il servizio che corrisponde all'ARN specificato in Simulation Settings (Impostazioni simulazione).
Questa operazione appartiene a un servizio che supporta speciali meccanismi di controllo degli accessi, oltre a policy basate su risorse, ad esempio ACL Amazon S3 o policy Vault Lock S3 Glacier. Il simulatore di policy non supporta questi meccanismi, perciò i risultati possono variare in base all'ambiente di produzione.	Nessuna operazione necessaria. Questo messaggio è informativo. Nella versione corrente, il simulatore valuta le policy collegate a utenti e gruppi di utenti; è inoltre in grado di valutare le policy basate su risorse per Amazon S3, Amazon SQS, Amazon SNS e S3 Glacier. Il simulatore di policy non supporta tutti i meccanismi di controllo degli accessi supportati da altri servizi AWS .

Messaggio	Procedura per la risoluzione
DynamoDB FGAC attualmente non è supportata.	Nessuna operazione necessaria. Questo messaggio informativo si riferisce a un controllo granulare degli accessi. Il controllo granulare degli accessi è la possibilità di utilizzare le condizioni delle policy IAM per determinare chi può accedere a singoli elementi di dati e attributi nelle tabelle e negli indici DynamoDB. Si riferisce anche alle azioni che possono essere eseguite su queste tabelle e indici. La versione corrente del simulatore di policy IAM non supporta questo tipo di condizione di policy. Per ulteriori informazioni sul controllo granulare degli accessi a DynamoDB, consulta Controllo granulare degli accessi per DynamoDB.
Si dispone di policy che non rispettano la sintassi della policy. È possibile utilizzare il validatore della policy per rivedere gli aggiornamenti consigliati per le policy.	Questo messaggio viene visualizzato nella parte superiore dell'elenco della policy se si dispone di policy che non sono conformi alla sintassi delle policy IAM. Per simulare queste policy, consultare le opzioni di convalida delle policy all'indirizzo Convalida delle policy IAM per identificare e correggere le policy.
Questa policy deve essere aggiornata per essere conforme alle regole più recenti della sintassi della policy.	Questo messaggio viene visualizzato se si dispone di policy che non sono conformi alla grammatica della policy IAM. Per simulare queste policy, consultare le opzioni di convalida delle policy all'indirizzo Convalida delle policy IAM per identificare e correggere le policy.

Utilizzo del simulatore di policy IAM (e dell'API)AWS CLIAWS

I comandi del simulatore di policy richiedono solitamente il richiamo delle operazioni API per eseguire due operazioni:

1. Valutare le policy e restituire l'elenco delle chiavi di contesto alle quali fanno riferimento. È necessario sapere a quali chiavi contestuali viene fatto riferimento in modo da potervi fornire valori nella fase successiva.
2. Simulare le policy, fornendo un elenco di operazioni, risorse e chiavi di contesto utilizzate durante la simulazione.

Per motivi di sicurezza, le operazioni API sono state suddivise in due gruppi:

- Le operazioni API che simulano solo le policy che vengono passate direttamente all'API come stringhe. Questo set include GetContextKeysForCustomPolicye. SimulateCustomPolicy
- Le operazioni API che simulano le policy che sono collegate a un utente, gruppo di utenti, ruolo o risorsa IAM specifici. Poiché queste operazioni API possono rivelare i dettagli delle autorizzazioni assegnate ad altre entità IAM, è consigliabile limitare l'accesso a queste operazioni API. Questo set include GetContextKeysForPrincipalPolicye SimulatePrincipalPolicy. Per ulteriori informazioni su come limitare l'accesso alle operazioni API;, consultare Politiche di esempio: AWS Identity and Access Management (IAM).

In entrambi i casi, le operazioni API simulano l'effetto di una o più policy su un elenco di operazioni e risorse. Ogni operazione è associata a ciascuna risorsa e la simulazione determina se le policy permettono o negano l'operazione per quella risorsa. È anche possibile fornire i valori per chiavi di contesto alle quali fanno riferimento le policy. È possibile ottenere l'elenco delle chiavi di contesto alle quali fanno riferimento le policy chiamando prima `GetContextKeysForCustomPolicy` o `GetContextKeysForPrincipalPolicy`. Se non si fornisce un valore per una chiave di contesto, la simulazione viene ancora eseguita. Tuttavia, i risultati potrebbero non essere affidabili, perché il simulatore non può includere quella chiave di contesto nella valutazione.

Per ottenere l'elenco delle chiavi contestuali (AWS CLI, AWS API)

Utilizzare quanto segue per valutare un elenco di policy e restituire un elenco di chiavi di contesto utilizzate nella policy.

- AWS CLI: `aws iam get-context-keys-for-custom-policy` e `aws iam get-context-keys-for-principal-policy`

- AWS API: `GetContextKeysForCustomPolicy`e `GetContextKeysForPrincipalPolicy`

Per simulare le politiche IAM (AWS CLI, AWS API)

Utilizza quanto segue per simulare le policy IAM per determinare le autorizzazioni valide di un utente.

- AWS CLI: `aws iam simulate-custom-policy` e `aws iam simulate-principal-policy`
- AWS API: `SimulateCustomPolicy`e `SimulatePrincipalPolicy`

Aggiunta e rimozione di autorizzazioni per identità IAM

Puoi utilizzare le policy per definire le autorizzazioni per una identità (utente, gruppo di utenti o ruolo). Puoi aggiungere e rimuovere le autorizzazioni allegando e scollegando le policy IAM per un'identità utilizzando AWS Management Console, the AWS Command Line Interface (AWS CLI) o l'API. AWS Puoi usare le policy anche per impostare limiti delle autorizzazioni solo per le entità (utenti o ruoli) che utilizzano gli stessi metodi. I limiti delle autorizzazioni sono una AWS funzionalità avanzata che controlla il numero massimo di autorizzazioni che un'entità può avere.

Argomenti

- Terminologia
- Visualizzazione dell'attività delle identità
- Aggiunta di autorizzazioni per identità IAM (console)
- Rimozione delle autorizzazioni per le identità IAM (console)
- Aggiunta di policy IAM (AWS CLI)
- Rimozione di policy IAM (AWS CLI)
- Aggiungere politiche IAM (AWS API)
- Rimozione delle politiche IAM (API)AWS

Terminologia

Quando associ le policy di autorizzazione alle identità (utenti, gruppi di utenti e ruoli), la terminologia e le procedure variano a seconda che lavori con una policy gestita o con una policy in linea:

- Collega: utilizzato con le policy gestite. Una policy gestita si collega a un'identità (utente, gruppo di utenti o ruolo). Il collegamento di una policy prevede l'applicazione delle sue autorizzazioni all'identità.

- Distacca: utilizzato con le policy gestite. Una policy gestita viene scollegata da un'identità IAM (utente, gruppo di utenti o ruolo). Il distaccamento di una policy prevede la rimozione delle sue autorizzazioni dall'identità.
- Integra: utilizzato con le policy in linea. Una policy in linea viene integrata in una identità (utente, gruppo di utenti o ruolo). L'incorporamento di un policy prevede l'applicazione delle sue autorizzazioni all'identità. Poiché una policy inline è memorizzata nell'identità, è incorporata anziché collegata, anche se i risultati sono simili.

> **Note**
>
> Puoi incorporare una policy inline per un ruolo collegato al servizio solo in un servizio che dipende dal ruolo. Consulta la Documentazione di AWS relativa al servizio per scoprire se questa funzionalità è supportata.

- Elimina: utilizzato con le policy in linea. Una policy in linea viene eliminata da un'identità IAM (utente, gruppo di utenti o ruolo). L'eliminazione di una policy rimuove le sue autorizzazioni dall'identità.

> **Note**
>
> Puoi eliminare una policy inline per un ruolo collegato al servizio solo in un servizio che dipende dal ruolo. Consulta la Documentazione di AWS relativa al servizio per scoprire se questa funzionalità è supportata.

Puoi utilizzare la console o l' AWS CLI AWS API per eseguire una qualsiasi di queste azioni.

Ulteriori informazioni

- Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.
- Per ulteriori informazioni sui limiti delle autorizzazioni, consultare Limiti delle autorizzazioni per le entità IAM.
- Per informazioni generali sulle policy IAM, consulta Policy e autorizzazioni in IAM.
- Per ulteriori informazioni sulla convalida delle policy IAM, consulta Convalida delle policy IAM.
- Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Visualizzazione dell'attività delle identità

Prima di modificare le autorizzazioni per un'identità (utente, gruppo di utenti o ruolo), è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Aggiunta di autorizzazioni per identità IAM (console)

Puoi utilizzare il AWS Management Console per aggiungere autorizzazioni a un'identità (utente, gruppo di utenti o ruolo). A tale scopo, collega le policy gestite che controllano le autorizzazioni oppure specifica una policy che funga da limite delle autorizzazioni. È inoltre possibile incorporare una policy inline.

Per usare una policy gestita come policy di autorizzazione per un'identità (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco di policy seleziona il pulsante di opzione accanto al nome della policy da collegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Scegli Operazioni e seleziona Collega.
5. Seleziona una o più identità per collegarle alla policy. Puoi usare la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver selezionato le identità, scegliere Attach policy (Collega policy).

Per usare una policy gestita per impostare un limite delle autorizzazioni (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.

4. Nella pagina dei dettagli della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni e seleziona Imposta la policy come limite delle autorizzazioni.
5. Selezionare uno o più utenti o ruoli su cui utilizzare la policy per un limite delle autorizzazioni. Puoi usare la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver selezionato i principali, scegli Imposta limite autorizzazioni.

Per incorporare una policy inline per un utente o un ruolo (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, selezionare Users (Utenti) o Roles (Ruoli).
3. Nell'elenco, scegli il nome dell'utente o il ruolo in cui incorporare una policy.
4. Scegli la scheda Autorizzazioni.
5. Scegli Aggiungi autorizzazioni, quindi seleziona Crea policy inline.

> **Note**
>
> Non è possibile integrare una policy in linea in un ruolo collegato ai servizi in IAM. Poiché il servizio collegato definisce se puoi modificare le autorizzazioni del ruolo, potresti aggiungere ulteriori policy dalla console di servizio, dall'API o dall' AWS CLI. Per visualizzare la documentazione del ruolo collegato al servizio per un servizio, consulta la pagina AWS servizi che funzionano con IAM e scegli Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato al servizio) per il servizio.

6. Scegli tra i seguenti metodi per visualizzare i passaggi necessari per creare le tue policy:
   - L'importazione di policy gestite esistenti. È possibile importare una policy gestita all'interno del proprio account e quindi modificare la policy per personalizzarla in base a requisiti specifici. Una policy gestita può essere una policy AWS gestita o una policy gestita dal cliente che hai creato in precedenza.
   - Creazione di policy con l'editor visivo. È possibile creare una nuova policy da zero nell'editor visivo. Se si utilizza l'editor visivo, non è necessario comprendere la sintassi JSON.

- Creazione di policy utilizzando l'editor JSON: nell'opzione dell'editor JSON, puoi utilizzare la sintassi JSON per creare una policy. È possibile scrivere un nuovo documento di policy JSON o incollare un esempio di policy.

7. Una volta creata, una policy inline viene automaticamente incorporata all'utente o al ruolo.

Come integrare una policy in linea per un gruppo di utenti (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Gruppi di utenti.
3. Nell'elenco, scegli il nome del gruppo di utenti in cui integrare una policy.
4. Seleziona la scheda Autorizzazioni, quindi Aggiungi autorizzazioni e Crea policy in linea.
5. Esegui una di queste operazioni:
   - Seleziona l'opzione Visivo per creare la policy. Per ulteriori informazioni, consulta Creazione di policy con l'editor visivo.
   - Scegli l'opzione JSON per creare la policy. Per ulteriori informazioni, consulta Creazione di policy utilizzando l'editor JSON.
6. Al termine, scegliere Create policy (Crea policy).

Per modificare il limite delle autorizzazioni per una o più entità (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Nella pagina dei dettagli della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni. Seleziona la casella di controllo accanto agli utenti o ai ruoli i cui limiti devono essere modificati e scegli Modifica.
5. Selezionare una nuova policy per l'utilizzo di un limite delle autorizzazioni. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy. Dopo aver selezionato la policy, scegli Imposta limite autorizzazioni.

Rimozione delle autorizzazioni per le identità IAM (console)

Puoi utilizzare il AWS Management Console per rimuovere le autorizzazioni da un'identità (utente, gruppo di utenti o ruolo). A tale scopo, scollega le policy gestite che controllano le autorizzazioni oppure rimuovi la policy che funge da limite delle autorizzazioni. È inoltre possibile eliminare una policy inline.

Per distaccare una policy gestita utilizzata come policy di autorizzazione (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco di policy seleziona il pulsante di opzione accanto al nome della policy da scollegare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni, quindi Distacca.
5. Seleziona le identità da distaccare dalla policy. È possibile utilizzare la casella di ricerca per filtrare l'elenco di identità. Dopo aver selezionato le identità, scegliere Detach policy (Scollega policy).

Per rimuovere un limite delle autorizzazioni (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, scegliere il nome della policy da impostare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Nella pagina di riepilogo della policy, scegli la scheda Entità collegate e quindi, se necessario, apri la sezione Collega come limite delle autorizzazioni e scegli le entità da cui rimuovere i limiti delle autorizzazioni. Quindi scegli Rimuovi limite.
5. Confermare la rimozione del limite e scegli Rimuovi limite.

Per eliminare una policy inline (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
3. Nell'elenco, scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da rimuovere.
4. Scegli la scheda Autorizzazioni.
5. Seleziona la casella di controllo accanto alla policy e scegli Rimuovi.
6. Nella finestra di dialogo di conferma seleziona Rimuovi.

Aggiunta di policy IAM (AWS CLI)

Puoi utilizzare il AWS CLI per aggiungere autorizzazioni a un'identità (utente, gruppo di utenti o ruolo). A tale scopo, collega le policy gestite che controllano le autorizzazioni oppure specifica una policy che funga da limite delle autorizzazioni. È inoltre possibile incorporare una policy inline.

Per usare una policy gestita come policy di autorizzazione per un'entità (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: get-policy
2. Per collegare una policy gestita a un'identità (utente, gruppo di utenti o ruolo), utilizza uno dei seguenti comandi:
   - come sono attach-user-policy
   - era io attach-group-policy
   - era io attach-role-policy

Per usare una policy gestita per impostare un limite delle autorizzazioni (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy
2. Per usare una policy gestita per impostare il limite delle autorizzazioni per un'entità (utente o ruolo), utilizzare uno dei comandi seguenti:
   - era io put-user-permissions-boundary
   - era io put-role-permissions-boundary

Per incorporare una policy inline (AWS CLI)

Per integrare una policy in linea in un'identità (utente, gruppo o ruolo che non è un ruolo collegato ai servizi), utilizza uno dei seguenti comandi:

- era io put-user-policy
- era io put-group-policy
- era io put-role-policy

Rimozione di policy IAM (AWS CLI)

È possibile utilizzare il AWS CLI per scollegare le politiche gestite che controllano le autorizzazioni o rimuovere una politica che funge da limite delle autorizzazioni. È inoltre possibile eliminare una policy inline.

Per distaccare una policy gestita utilizzata come policy di autorizzazione (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:
   - Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita:
     - era io list-entities-for-policy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), usa uno dei comandi seguenti:
     - era io list-attached-user-policies
     - era io list-attached-group-policies
     - era io list-attached-role-policies
3. Per distaccare una policy gestita da un'identità (utente, gruppo di utenti o ruolo), utilizza uno dei seguenti comandi:
   - era io detach-user-policy
   - era io detach-group-policy
   - era io detach-role-policy

Per rimuovere un limite delle autorizzazioni (AWS CLI)

1. (Facoltativo) Per visualizzare la policy gestita attualmente utilizzata per impostare il limite delle autorizzazioni per un utente o ruolo, eseguire i comandi seguenti:
   - aws iam get-user
   - aws iam get-role
2. (Facoltativo) Per visualizzare gli utenti o i ruoli su cui si utilizza una policy gestita per un limite delle autorizzazioni, eseguire il comando seguente:
   - era io list-entities-for-policy
3. (Facoltativo) Per visualizzare le informazioni su una policy gestita, eseguire i comandi seguenti:
   - Per elencare le policy gestite: aws iam list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: aws iam get-policy
4. Per rimuovere un limite delle autorizzazioni da un utente o ruolo, utilizzare uno dei comandi seguenti:
   - era io delete-user-permissions-boundary
   - era io delete-role-permissions-boundary

Per eliminare una policy inline (AWS CLI)

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), usa uno dei seguenti comandi:
   - era io list-user-policies
   - era io list-group-policies
   - era io list-role-policies
2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), usa uno dei seguenti comandi:
   - era io get-user-policy
   - era io get-group-policy
   - era io get-role-policy

3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), usa uno dei seguenti comandi:

   - era io delete-user-policy
   - era io delete-group-policy
   - era io delete-role-policy

Aggiungere politiche IAM (AWS API)

Puoi utilizzare l' AWS API per allegare policy gestite che controllano le autorizzazioni o specificare una policy che funga da limite di autorizzazione. È inoltre possibile incorporare una policy inline.

Per utilizzare una politica gestita come politica di autorizzazioni per un'entità (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:

   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. Per collegare una policy gestita a un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:

   - AttachUserPolicy
   - AttachGroupPolicy
   - AttachRolePolicy

Per utilizzare una politica gestita per impostare un limite di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy gestita, chiamare le operazioni seguenti:

   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. Per usare una policy gestita per impostare il limite delle autorizzazioni per un'entità (utente o ruolo), chiamare una delle operazioni seguenti:

   - PutUserPermissionsBoundary
   - PutRolePermissionsBoundary

Per incorporare una policy in linea (API)AWS

Per integrare una policy in un'identità (utente, gruppo o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

- PutUserPolicy
- PutGroupPolicy
- PutRolePolicy

Rimozione delle politiche IAM (API)AWS

Puoi utilizzare l' AWS API per scollegare le policy gestite che controllano le autorizzazioni o rimuovere una policy che funge da limite di autorizzazione. È inoltre possibile eliminare una policy inline.

Per scollegare una politica gestita utilizzata come politica di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:
   - Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), chiama una delle operazioni seguenti:
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies
3. Per distaccare una policy gestita da un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:
   - DetachUserPolicy
   - DetachGroupPolicy
   - DetachRolePolicy

Per rimuovere un limite di autorizzazioni (API)AWS

1. (Facoltativo) Per visualizzare la policy gestita attualmente utilizzata per impostare il limite delle autorizzazioni per un utente o ruolo, chiamare le operazioni seguenti:
   - GetUser
   - GetRole
2. (Facoltativo) Per visualizzare gli utenti o i ruoli su cui si utilizza una policy gestita per un limite delle autorizzazioni, chiamare l'operazione seguente:
   - ListEntitiesForPolicy
3. (Facoltativo) Per visualizzare le informazioni su una policy gestita, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
4. Per rimuovere un limite delle autorizzazioni da un utente o ruolo, chiamare una delle operazioni seguenti:
   - DeleteUserPermissionsBoundary
   - DeleteRolePermissionsBoundary

Per eliminare una policy in linea (API)AWS

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), chiama una delle seguenti operazioni:
   - ListUserPolicies
   - ListGroupPolicies
   - ListRolePolicies
2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:
   - GetUserPolicy
   - GetGroupPolicy
   - GetRolePolicy

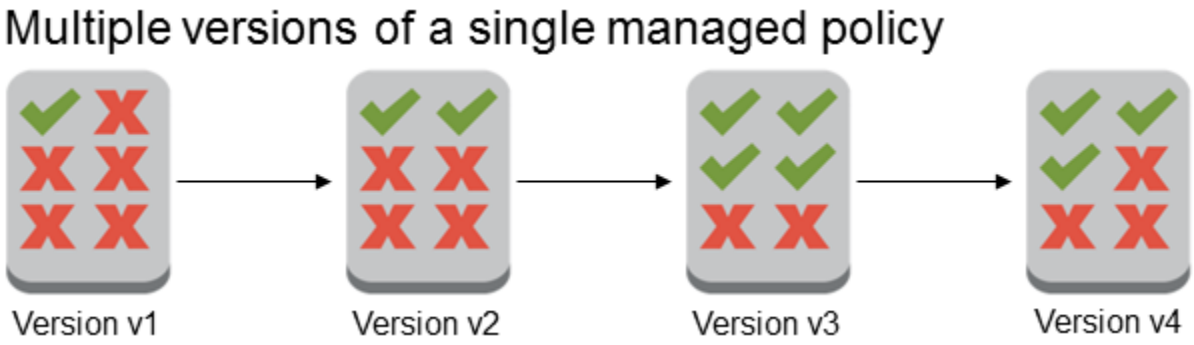
3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

   - DeleteUserPolicy
   - DeleteGroupPolicy
   - DeleteRolePolicy

Controllo delle versioni delle policy IAM

Quando apporti modifiche a una policy gestita dai clienti IAM e quando AWS apporti modifiche a una policy AWS gestita, la policy modificata non sovrascrive la policy esistente. IAM crea invece una nuova versione della policy gestita. IAM memorizza fino a cinque versioni di una policy gestita dal cliente. IAM non supporta il controllo delle versioni per le policy in linea.

Il diagramma seguente illustra la funzione Versioni multiple per una policy gestita dal cliente. In questo esempio, vengono salvate le versioni 1-4. Puoi salvare fino a cinque versioni delle policy gestite in IAM. Quando si modifica una policy che creerebbe una sesta versione salvata, è possibile scegliere quale versione precedente non memorizzare più. È possibile ripristinare una qualsiasi delle altre quattro versioni salvate in qualsiasi momento.



Una versione di policy è diversa da un elemento `Version` della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version.

È possibile utilizzare le versioni per tenere traccia delle modifiche apportate a una policy gestita. Ad esempio, è possibile effettuare una modifica a una policy gestita e quindi scoprire che la modifica ha avuto effetti non previsti. In questo caso, è possibile eseguire il rollback a una versione precedente della policy gestita impostando la versione precedente come versione predefinita.

Negli argomenti seguenti viene illustrato come è possibile utilizzare il controllo delle versioni per le policy gestite.

Argomenti

Autorizzazioni per impostare la versione predefinita di una policy

Le autorizzazioni necessarie per impostare la versione predefinita di una policy corrispondono alle operazioni API di AWS per l'attività. È possibile utilizzare l'operazione API `CreatePolicyVersion` o `SetDefaultPolicyVersion` per impostare la versione predefinita di una policy. Per consentire a un utente di impostare la versione predefinita di una policy esistente, è possibile consentire l'accesso all'operazione `iam:CreatePolicyVersion` o all'operazione `iam:SetDefaultPolicyVersion`. L'operazione `iam:CreatePolicyVersion` consente di creare una nuova versione della policy e di impostarla come predefinita. L'operazione `iam:SetDefaultPolicyVersion` consente di impostare qualsiasi versione esistente della policy come predefinita.

> **Important**
>
> Negare l'operazione `iam:SetDefaultPolicyVersion` nella policy di un utente non impedisce a quest'ultimo di creare una nuova versione della policy e di impostarla come predefinita.

È possibile utilizzare la policy seguente per negare a un utente l'accesso per modificare una policy gestita dal cliente esistente:

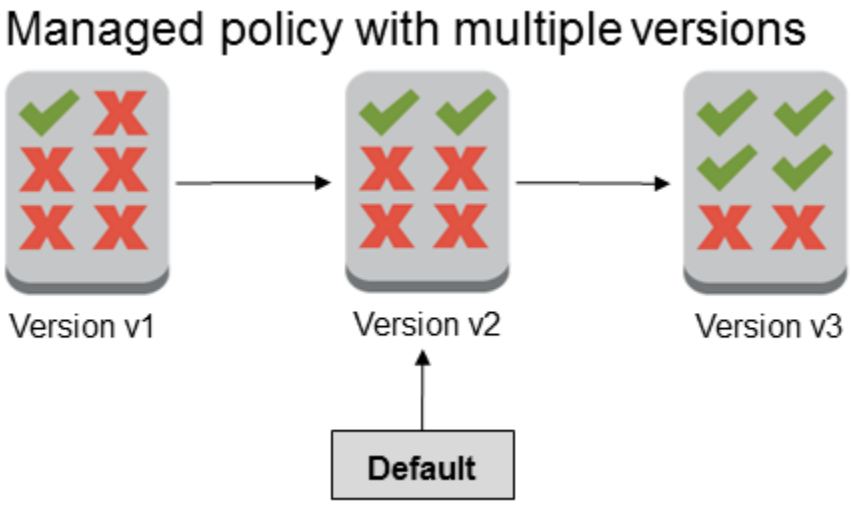
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "iam:CreatePolicyVersion",
```

```
                "iam:SetDefaultPolicyVersion"
            ],
            "Resource": "arn:aws:iam::*:policy/POLICY-NAME"
        }
    ]
}
```

Impostare la versione predefinita di una policy gestita dal cliente

Una delle versioni di una policy gestita viene impostata come versione predefinita. La versione di default della policy è la versione operativa, ovvero, è la versione valida per tutte le entità principali (utenti, gruppi di utenti e ruoli) a cui è collegata la policy gestita.

Quando si crea una policy gestita dal cliente, la policy inizia con una singola versione identificata come v1. Per le policy gestite con una sola versione, tale versione viene automaticamente impostata come predefinita. Per le policy gestite dal cliente con più di una versione, selezionare la versione da impostare come predefinita. Per le policy AWS gestite, la versione predefinita è impostata da AWS. I seguenti diagrammi illustrano questo concetto.



Puoi impostare la versione predefinita di una policy gestita dal cliente per applicarla a tutte le identità IAM (utente, gruppo di utenti e ruolo) a cui è collegata la policy. Non è possibile impostare la versione predefinita per una politica AWS gestita o una politica in linea.

Per impostare la versione predefinita di una policy gestita dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy per cui impostare la versione predefinita. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Selezionare la casella di controllo a fianco della versione da impostare come predefinita e selezionare Set as default (Imposta come predefinita).

Per informazioni su come impostare la versione predefinita di una policy gestita dal cliente dall'API AWS Command Line Interface o dall' AWS API, consultaModifica di policy gestite dal cliente (AWS CLI).

Utilizzo delle versioni per il rollback delle modifiche

È possibile impostare la versione predefinita di una policy gestita dal cliente per eseguire il rollback delle modifiche. Si consideri ad esempio lo scenario riportato di seguito:

Crea una policy gestita dal cliente che consenta agli utenti di amministrare un determinato bucket Amazon S3 utilizzando la AWS Management Console. Al momento della creazione, la policy gestita dal cliente ha solo una versione, identificata come v1, in modo che questa versione venga automaticamente impostata come predefinita. La policy funziona come previsto.

Successivamente, si aggiorna la policy per aggiungere l'autorizzazione per amministrare un secondo bucket Amazon S3. IAM crea una nuova versione della policy, identificata come v2, che contiene le modifiche. Imposta la versione v2 come predefinita e poco dopo gli utenti segnalano che non dispongono dell'autorizzazione per utilizzare la console Amazon S3. In questo caso, è possibile ripristinare la versione v1 della policy, che funziona come previsto. Per eseguire questa operazione, è necessario impostare la versione v1 come versione predefinita. Gli utenti sono ora in grado di utilizzare la console Amazon S3 per amministrare il bucket originale.

Successivamente, dopo aver determinato l'errore nella versione v2 della policy, aggiorna nuovamente la policy per aggiungere l'autorizzazione per amministrare il secondo bucket Amazon S3. IAM crea una nuova versione della policy, identificata come v3. Si imposta quindi la versione v3 come predefinita e questa versione funziona come previsto. A questo punto, si elimina la versione v2 della policy.

Limiti della versione

Una policy gestita può avere fino a cinque versioni. Se devi apportare modifiche a una policy gestita in più di cinque versioni della AWS Command Line Interface o dell' AWS API, devi prima eliminare una o più versioni esistenti. Se si utilizza la AWS Management Console, non è necessario eliminare una versione prima di modificare la politica. Quando si salva una sesta versione, verrà visualizzata una finestra di dialogo che richiede di eliminare una o più versioni non predefinite della policy. È possibile visualizzare il documento della policy JSON per ogni versione per facilitare la scelta. Per ulteriori informazioni su questa finestra di dialogo, consultare the section called “Modifica delle policy IAM”.

È possibile eliminare qualsiasi versione di policy gestita desiderata, ad eccezione della versione predefinita. Quando si elimina una versione, gli identificatori di versione per le versioni rimanenti non vengono modificati. Di conseguenza, gli identificativi di versione potrebbero non essere sequenziali. Ad esempio, se si eliminano le versioni v2 e v4 di una policy gestita e si aggiungono due nuove versioni, gli identificativi della versione rimanenti potrebbero essere v1, v3, v5, v6 e v7.

Modifica delle policy IAM

Una policy è un'entità che, se viene collegata a un'identità o a una risorsa, ne definisce le autorizzazioni. Le policy sono archiviate AWS come documenti JSON e sono allegate ai principali come policy basate sull'identità in IAM. Puoi collegare una policy basata sulle identità a un principale (o identità), ad esempio un gruppo, un utente o un ruolo IAM. Le politiche basate sull'identità includono politiche gestite, politiche AWS gestite dai clienti e politiche in linea. Puoi modificare le politiche gestite dai clienti e le politiche in linea in IAM. AWS le politiche gestite non possono essere modificate. Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Argomenti

- Visualizzazione dell'accesso alle policy
- Modifica di policy gestite dal cliente (console)
- Modifica delle policy in linea (console)
- Modifica di policy gestite dal cliente (AWS CLI)
- Modifica delle politiche gestite dai clienti (AWS API)

Visualizzazione dell'accesso alle policy

Prima di modificare le autorizzazioni per una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Modifica di policy gestite dal cliente (console)

Puoi modificare le policy gestite dal cliente per cambiare le autorizzazioni in esse definite. Possono esistere fino a cinque versioni di una policy gestita dal cliente. Questo è importante perché se apporti modifiche a una policy gestita oltre cinque versioni, AWS Management Console ti viene richiesto di decidere quale versione eliminare. Per evitare questo problema, puoi selezionare di modificare la versione predefinita oppure eliminare una versione di una policy prima di apportare modifiche. Per ulteriori informazioni sulle versioni, consultare Controllo delle versioni delle policy IAM.

Per modificare una policy gestita dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy da modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Seleziona la scheda Autorizzazioni e scegli Modifica.
5. Esegui una di queste operazioni:

   - Per modificare la policy senza conoscere la sintassi JSON, seleziona l'opzione Visivo. Puoi modificare servizi, operazioni, risorse o condizioni opzionali per ogni blocco di autorizzazione della policy. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Al termine, seleziona Successivo per continuare.
   - Seleziona la scheda JSON per modificare la policy, digitando o copiando il testo nella casella JSON. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

6. Nella pagina Verifica e salva, esamina il campo Autorizzazioni definite in questa policy, quindi scegli Salva modifiche per salvare il lavoro.
7. Se esistono già un massimo di cinque versioni della policy gestita, seleziona Salva per visualizzare una finestra di dialogo. Per salvare la nuova versione, la versione non predefinita più vecchia della policy viene rimossa e sostituita con la nuova. Facoltativamente, puoi impostare la nuova versione come versione predefinita della policy.

   Scegli Salva modifiche per salvare la nuova versione della policy.

Per impostare la versione predefinita di una policy gestita dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy per cui impostare la versione predefinita. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Selezionare la casella di controllo a fianco della versione da impostare come predefinita e selezionare Set as default (Imposta come predefinita).

Per eliminare una versione di una policy gestita dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Selezionare il nome della policy gestita dal cliente di cui eliminare una versione. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle policy.
4. Selezionare la scheda Policy versions (Versioni policy). Selezionare la casella di controllo accanto alla versione da eliminare. Scegli Elimina.
5. Confermare l'eliminazione della versione e selezionare Delete (Elimina).

Modifica delle policy in linea (console)

Le policy inline possono essere modificate dalla AWS Management Console.

Come modificare una policy in linea per un utente, un gruppo di utenti o ruolo (console)

1. Nel pannello di navigazione scegli Gruppi, Utenti o Ruoli.
2. Scegli il nome dell'utente, del gruppo di utenti o del ruolo con la policy da modificare. Selezionare la scheda Permissions (Autorizzazioni) ed espandere la policy.
3. Per modificare una policy inline, selezionare Edit Policy (Modifica policy).
4. Esegui una di queste operazioni:

   - Per modificare la policy senza conoscere la sintassi JSON, seleziona l'opzione Visivo. Puoi modificare servizi, operazioni, risorse o condizioni opzionali per ogni blocco di autorizzazione

della policy. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Al termine, seleziona Successivo per continuare.

- Seleziona la scheda JSON per modificare la policy, digitando o copiando il testo nella casella JSON. Inoltre, puoi importare una policy per aggiungere ulteriori autorizzazioni nella parte finale. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Per salvare le modifiche senza influenzare le entità collegate al momento, deselezionare la casella di controllo Save as default version (Salva come versione predefinita).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

5. Nella pagina Rivedi consulta il riepilogo della policy e scegli Salva modifiche per salvare il lavoro.

Modifica di policy gestite dal cliente (AWS CLI)

Puoi modificare una policy gestita dai clienti da AWS Command Line Interface (AWS CLI).

> **Note**
>
> Una policy gestita può avere fino a cinque versioni. Per apportare modifiche a una policy gestita dal cliente di cui esistono già cinque versioni, devi eliminare una o più versioni esistenti.

Per modificare una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per elencare le policy gestite: list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: get-policy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:

- Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita:
  - list-entities-for-policy
- Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo):
  - list-attached-user-policies
  - list-attached-group-policies
  - list-attached-role-policies

3. Per modificare una policy gestita dal cliente, eseguire il comando seguente:
   - create-policy-version
4. (Facoltativo) Per convalidare una policy gestita dal cliente, esegui il comando IAM Access Analyzer seguente:
   - validate-policy

Per impostare la versione predefinita di una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per elencare le policy gestite, eseguire il comando seguente:
   - list-policies
2. Per impostare la versione predefinita di una policy gestita dal cliente, eseguire il comando seguente:
   - set-default-policy-version

Per eliminare una versione di una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per elencare le policy gestite, eseguire il comando seguente:
   - list-policies
2. Per eliminare una policy gestita dal cliente, eseguire il comando seguente:
   - delete-policy-version

Modifica delle politiche gestite dai clienti (AWS API)

Puoi modificare una politica gestita dai clienti utilizzando l' AWS API.

> **Note**
>
> Una policy gestita può avere fino a cinque versioni. Per apportare modifiche a una policy gestita dal cliente di cui esistono già cinque versioni, devi eliminare una o più versioni esistenti.

Per modificare una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:
   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:
   - Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo):
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies
3. Per modificare una policy gestita dal cliente, richiamare la seguente operazione:
   - CreatePolicyVersion
4. (Facoltativo) Per convalidare una policy gestita dal cliente, richiama la seguente operazione IAM Access Analyzer:
   - ValidatePolicy

Per impostare la versione predefinita di una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per elencare le policy gestite, richiamare la seguente operazione:
   - ListPolicies
2. Per impostare la versione predefinita di una policy gestita dal cliente, richiamare la seguente operazione:

- SetDefaultPolicyVersion

Per eliminare una versione di una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per elencare le policy gestite, richiamare la seguente operazione:
   - ListPolicies
2. Per eliminare una policy gestita dal cliente, chiamare l'operazione seguente:
   - DeletePolicyVersion

Eliminazione di policy IAM

Puoi eliminare le policy IAM utilizzando AWS Management Console, the AWS Command Line Interface (AWS CLI) o l'API IAM.

> **Note**
>
> L'eliminazione delle policy IAM è definitiva. Una volta eliminata, la policy non potrà più essere ripristinata.

Per ulteriori informazioni sulle differenze tra policy gestite e policy inline, consulta Policy gestite e policy inline.

Per informazioni generali sulle policy IAM, consulta Policy e autorizzazioni in IAM.

Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

Argomenti

- Visualizzazione dell'accesso alle policy
- Eliminazione di policy IAM (console)
- Eliminazione di policy IAM (AWS CLI)
- Eliminazione delle politiche IAM (AWS API)

Visualizzazione dell'accesso alle policy

Prima di eliminare una policy, è opportuno esaminare la sua attività recente a livello di servizio. È un'opzione importante per non rimuovere l'accesso da parte di un principale (persona o applicazione) che la sta utilizzando. Per ulteriori informazioni sulla visualizzazione delle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Eliminazione di policy IAM (console)

Puoi eliminare una policy gestita dal cliente per rimuoverla dal tuo Account AWS. Non è possibile eliminare le policy AWS gestite.

Per eliminare una policy gestita dal cliente (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Seleziona il pulsante di opzione accanto alla policy gestita dal cliente da eliminare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Azioni, quindi Elimina.
5. Segui le istruzioni per confermare che desideri eliminare la policy, quindi scegli Elimina.

Come eliminare una policy in linea per un gruppo di utenti, un utente o un ruolo (console)

1. Nel pannello di navigazione scegli Gruppi di utenti, Utenti o Ruoli.
2. Scegli il nome del gruppo di utenti, dell'utente o del ruolo con la policy da eliminare. Selezionare la scheda Permissions (Autorizzazioni).
3. Seleziona le caselle di controllo accanto alle policy da eliminare, quindi scegli Rimuovi. Per eliminare una policy in linea in Utenti o Ruoli, scegli Rimuovi per confermare l'eliminazione. Se elimini una singola policy in linea in Gruppi di utenti digita il nome della policy e scegli Elimina. Se elimini più policy in linea in Gruppi di utenti, digita il numero di policy da eliminare seguito da **`inline policies`** e scegli Elimina. Ad esempio, se elimini tre policy in linea, digita **`3 inline policies`**.

Eliminazione di policy IAM (AWS CLI)

Puoi eliminare una policy gestita dal cliente dall' AWS Command Line Interface.

Per eliminare una policy gestita dal cliente (AWS CLI)

1. (Facoltativo) Per visualizzare le informazioni su una policy, eseguire i comandi seguenti:
   - Per elencare le policy gestite: list-policies
   - Per recuperare informazioni dettagliate su una policy gestita: get-policy
2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, eseguire i comandi seguenti:
   - Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita, esegui il comando riportato sotto:
     - list-entities-for-policy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), esegui uno dei comandi riportati sotto:
     - list-attached-user-policies
     - list-attached-group-policies
     - list-attached-role-policies
3. Per eliminare una policy gestita dal cliente, eseguire il comando seguente:
   - delete-policy

Per eliminare una policy inline (AWS CLI)

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), usa uno dei seguenti comandi:
   - era iam list-user-policies
   - era io list-group-policies
   - era io list-role-policies
2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), usa uno dei seguenti comandi:
   - era io get-user-policy
   - era io get-group-policy

- era io get-role-policy

3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), usa uno dei seguenti comandi:

   - era io delete-user-policy
   - era io delete-group-policy
   - era io delete-role-policy

Eliminazione delle politiche IAM (AWS API)

Puoi eliminare una policy gestita dai clienti utilizzando l' AWS API.

Per eliminare una politica gestita dal cliente (AWS API)

1. (Facoltativo) Per visualizzare le informazioni su una policy, chiamare le operazioni seguenti:

   - Per elencare le politiche gestite: ListPolicies
   - Per recuperare informazioni dettagliate su una politica gestita: GetPolicy

2. (Facoltativo) Per scoprire le relazioni tra le policy e le identità, chiamare le operazioni seguenti:

   - Per elencare le identità (utenti, gruppi di utenti e ruoli) a cui è collegata una policy gestita, chiama la seguente operazione:
     - ListEntitiesForPolicy
   - Per elencare le policy gestite collegate a un'identità (utente, gruppo di utenti o ruolo), chiama una delle operazioni seguenti:
     - ListAttachedUserPolicies
     - ListAttachedGroupPolicies
     - ListAttachedRolePolicies

3. Per eliminare una policy gestita dal cliente, chiamare l'operazione seguente:

   - DeletePolicy

Per eliminare una policy in linea (API)AWS

1. (Opzionale) Per elencare tutte le policy in linea che sono collegate a un'identità (utente, gruppo di utenti, ruolo), chiama una delle seguenti operazioni:

- ListUserPolicies
- ListGroupPolicies
- ListRolePolicies

2. (Opzionale) Per recuperare un documento di policy in linea che è integrato in un'identità (utente, gruppo di utenti o ruolo), chiama una delle seguenti operazioni:

   - GetUserPolicy
   - GetGroupPolicy
   - GetRolePolicy

3. Per eliminare una policy in linea da un'identità (utente, gruppo di utenti o ruolo che non è un ruolo collegato ai servizi), chiama una delle seguenti operazioni:

   - DeleteUserPolicy
   - DeleteGroupPolicy
   - DeleteRolePolicy

Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso

In qualità di amministratore, puoi concedere alle risorse IAM (utenti, ruoli, gruppi di utenti o policy) autorizzazioni aggiuntive rispetto a quelle indispensabili. IAM fornisce le informazioni sull'ultimo accesso per facilitare l'identificazione delle autorizzazioni inutilizzate in modo da poterle rimuovere. È possibile utilizzare le informazioni relative all'ultimo accesso per perfezionare le policy e consentire l'accesso solo ai servizi e alle operazioni utilizzati dalle identità IAM. In questo modo è più facile rispettare le best practice dei privilegi minimi. È possibile visualizzare le informazioni relative all'ultimo accesso per le identità o le policy esistenti in IAM o AWS Organizations.

È possibile monitorare continuamente le informazioni relative all'ultimo accesso con analizzatori degli accessi inutilizzati. Per ulteriori informazioni, consulta Risultati relativi agli accessi esterni e inutilizzati.

Argomenti

- Tipi di informazioni sull'ultimo accesso per IAM
- Ultime informazioni di accesso per AWS Organizations
- Cose da sapere sulle ultime informazioni di accesso

- Autorizzazioni richieste
- Risoluzione dei problemi delle attività per le entità IAM e Organizations
- Dove AWS tiene traccia delle ultime informazioni a cui si accede
- Visualizzazione delle informazioni sull'ultimo accesso per IAM
- Visualizzazione delle informazioni sull'ultimo accesso per Organizations
- Scenari di esempio per l'utilizzo delle ultime informazioni di accesso
- Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM

Tipi di informazioni sull'ultimo accesso per IAM

È possibile visualizzare due tipi di informazioni relative all'ultimo accesso per le identità IAM: informazioni sui servizi AWS consentiti e informazioni sulle operazioni consentite. Le informazioni includono la data e l'ora in cui è stato effettuato il tentativo di accedere a un' AWS API. Per le operazioni, le informazioni relative all'ultimo accesso riportano le operazioni di gestione del servizio. Le azioni di gestione includono le azioni di creazione, eliminazione e modifica. Per ulteriori informazioni su come visualizzare le informazioni sull'ultimo accesso per IAM, consulta Visualizzazione delle informazioni sull'ultimo accesso per IAM.

Per esempi di scenari di impiego delle informazioni relative all'ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle identità IAM, consulta la pagina Scenari di esempio per l'utilizzo delle ultime informazioni di accesso.

Per ulteriori informazioni su come vengono fornite le informazioni per le azioni di gestione, vedere Cose da sapere sulle ultime informazioni di accesso.

Ultime informazioni di accesso per AWS Organizations

Se accedi utilizzando le credenziali dell'account di gestione, puoi visualizzare le informazioni sull'ultimo accesso al servizio per un' AWS Organizations entità o una politica dell'organizzazione. AWS Organizations le entità includono la radice dell'organizzazione, le unità organizzative (OU) o gli account. Le informazioni relative all'ultimo accesso AWS Organizations includono informazioni sui servizi consentiti da una policy di controllo dei servizi (SCP). Le informazioni indicano quali principali (utente root, ruolo o utente IAM) di un'organizzazione o un account hanno tentato l'ultimo accesso al servizio e quando. Per ulteriori informazioni sul rapporto e su come visualizzare le informazioni relative all'ultimo accesso AWS Organizations, vedereVisualizzazione delle informazioni sull'ultimo accesso per Organizations.

Per esempi di scenari dell'utilizzo delle informazioni sull'ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle entità di Organizations, consulta Scenari di esempio per l'utilizzo delle ultime informazioni di accesso.

Cose da sapere sulle ultime informazioni di accesso

Prima di utilizzare le informazioni relative all'ultimo accesso presenti in un report per modificare le autorizzazioni per un'identità IAM o un'entità Organizations, esamina i seguenti dettagli sulle informazioni.

- Periodo di monitoraggio: l'attività recente viene visualizzata nella console IAM entro quattro ore. Il periodo di monitoraggio per le informazioni sul servizio dura almeno 400 giorni, a seconda di quando il servizio ha avviato il monitoraggio delle informazioni sulle operazioni. Il periodo di monitoraggio delle informazioni sulle operazioni Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio per le operazioni di Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per tutti gli altri servizi è iniziato il 23 maggio 2023. Per un elenco dei servizi per i quali sono disponibili le informazioni relative all'ultimo accesso a un'operazione, consulta la pagina Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM. Per ulteriori informazioni sulle regioni per le quali sono disponibili le informazioni relative all'ultimo accesso a un'operazione, consulta la pagina Dove AWS tiene traccia delle ultime informazioni a cui si accede.
- Tentativi segnalati: i dati dell'ultimo accesso al servizio includono tutti i tentativi di accesso a un' AWS API, non solo quelli riusciti. Ciò include tutti i tentativi effettuati utilizzando l' AWS Management Console AWS API tramite uno qualsiasi degli SDK o uno qualsiasi degli strumenti a riga di comando. Una voce non prevista nell'ultimo accesso ai dati del servizio non significa che l'account è stato compromesso, poiché la richiesta potrebbe essere stata rifiutata. Fai riferimento ai tuoi CloudTrail log come fonte autorevole per informazioni su tutte le chiamate API e sull'esito positivo o negativo dell'accesso.
- PassRole— L'`iam:PassRole`azione non viene tracciata e non è inclusa nelle informazioni relative all'ultimo accesso dell'azione IAM.
- Informazioni sull'ultimo accesso all'operazione: le informazioni sull'ultimo accesso a un'operazione sono disponibili per le operazioni di gestione del servizio alle quali le identità IAM hanno eseguito l'accesso. Consulta l'elenco di servizi e delle relative operazioni per scoprire a quale operazione si riferiscono i report relativi all'ultimo accesso.

> **Note**
>
> Le informazioni sull'ultima operazione alla quale è stato effettuato l'accesso non sono disponibili per gli eventi di dati di Amazon S3.

- Eventi di gestione: IAM fornisce informazioni sulle azioni per gli eventi di gestione dei servizi registrati da. CloudTrail A volte, gli eventi di CloudTrail gestione vengono anche chiamati operazioni del piano di controllo o eventi del piano di controllo. Gli eventi di gestione forniscono visibilità sulle operazioni amministrative eseguite sulle risorse dell'azienda Account AWS. Per ulteriori informazioni sugli eventi di gestione in CloudTrail, vedere Registrazione degli eventi di gestione nella Guida per l'AWS CloudTrail utente.
- Proprietario del report: solo il principale che genera un report può visualizzare i dettagli del report. Ciò significa che quando si visualizzano le informazioni contenute in AWS Management Console, potrebbe essere necessario attendere che vengano generate e caricate. Se utilizzi l' AWS API AWS CLI or per ottenere i dettagli del report, le tue credenziali devono corrispondere alle credenziali del responsabile che ha generato il rapporto. Se si utilizzano credenziali temporanee per un ruolo o un utente federato, è necessario generare e recuperare il report durante la stessa sessione. Per ulteriori informazioni sulle entità principal di sessione del ruolo assunto, consulta AWS Elementi della policy JSON: Principal.
- Risorse IAM : le informazioni relative all'ultimo accesso per IAM includono le risorse IAM (ruoli, utenti, gruppi di utenti e policy) presenti nell'account. Le ultime informazioni a cui si accede per Organizations includono i principali (utenti IAM, ruoli IAM o i Utente root dell'account AWS) nell'entità Organizations specificata. Le informazioni relative all'ultimo accesso non includono i tentativi non autenticati.
- Tipi di policy IAM: le informazioni relative all'ultimo accesso per IAM includono i servizi consentiti dalle policy di un'identità IAM. Si tratta delle policy collegate a un ruolo o a un utente direttamente o tramite un gruppo. L'accesso consentito da altri tipi di policy non è incluso nel report. I tipi di policy esclusi includono le policy basate sulle risorse, le liste di controllo accessi, le SCP di AWS Organizations , i limiti delle autorizzazioni IAM e le policy di sessione. Le autorizzazioni fornite dai ruoli collegati ai servizi sono definite dal servizio a cui sono collegati e non possono essere modificati in IAM. Per ulteriori informazioni sui ruoli collegati ai servizi, vedere Uso di ruoli collegati ai servizi. Per informazioni sulla valutazione dei diversi tipi di criteri per consentire o negare l'accesso, vedere Logica di valutazione delle policy.
- Tipi di policy di Organizations: le informazioni per AWS Organizations includono solo servizi consentiti da una policy di controllo dei servizi (SCP) ereditata di un'entità Organizations. Le SCP

sono policy collegate a una root, una UO o un account L'accesso consentito da altri tipi di policy non è incluso nel report. I tipi di policy esclusi includono le policy basate sulle risorse, le liste di controllo accessi, i limiti delle autorizzazioni IAM e le policy di sessione. Per ulteriori informazioni su come i diversi tipi di policy vengono valutati per consentire o negare l'accesso, consulta Logica di valutazione delle policy.

- Specificazione di un ID di policy: quando si utilizza l' AWS API AWS CLI or per generare un report per le informazioni dell'ultimo accesso in Organizations, è possibile specificare facoltativamente un ID di policy. Il report risultante include i dati per i servizi consentiti solo da tale policy. Le informazioni includono l'attività più recente dell'account nell'entità Organizations specificata o nei relativi figli. Per ulteriori informazioni, consulta aws iam generate-organizations-access-report o. GenerateOrganizationsAccessReport
- Account di gestione di Organizations: è necessario accedere all'account di gestione dell'organizzazione per visualizzare le informazioni sull'ultimo accesso al servizio. Puoi scegliere di visualizzare le informazioni per l'account di gestione utilizzando la console IAM AWS CLI, l'o l' AWS API. Il report risultante elenca tutti i AWS servizi, poiché l'account di gestione non è limitato dagli SCP. Se specifichi un ID policy nella CLI o nell'API, la policy viene ignorata. Per ogni servizio, il report include le informazioni solo per l'account di gestione. Tuttavia, i report per altre entità di Organizations non riportano i dati delle attività nell'account di gestione.
- Impostazioni di Organizations: prima di poter generare i dati per Organizations, un amministratore deve abilitare le SCP nella tua root dell'organizzazione.

Autorizzazioni richieste

Per visualizzare le ultime informazioni a cui si accede in AWS Management Console, è necessario disporre di una politica che conceda le autorizzazioni necessarie.

Autorizzazioni per le informazioni IAM

Per utilizzare la console IAM per visualizzare le informazioni sull'ultimo accesso per un utente, ruolo o policy IAM, devi disporre di una policy che includa le seguenti operazioni:

- `iam:GenerateServiceLastAccessedDetails`
- `iam:Get*`
- `iam:List*`

Queste autorizzazioni consentono a un utente di visualizzare ciò che segue:

- Gli utenti, i gruppi o i ruoli collegati a una policy gestita
- I servizi cui può accedere un utente o ruolo
- L'ultima volta che ha effettuato l'accesso al servizio
- L'ultima volta che hanno provato a utilizzare un'operazione Amazon EC2, IAM, Lambda o Amazon S3 specifica

Per utilizzare l' AWS API AWS CLI or per visualizzare le informazioni dell'ultimo accesso per IAM, devi disporre delle autorizzazioni corrispondenti all'operazione che desideri utilizzare:

- `iam:GenerateServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetailsWithEntities`
- `iam:ListPoliciesGrantingServiceAccess`

Questo esempio mostra come creare una policy basata sull'identità che consenta di visualizzare le informazioni sull'ultimo accesso a IAM. Inoltre, consente l'accesso in sola lettura a tutto IAM. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateServiceLastAccessedDetails",
            "iam:Get*",
            "iam:List*"
        ],
        "Resource": "*"
    }
```

Autorizzazioni per le informazioni AWS Organizations

Per utilizzare la console IAM per visualizzare un report per la root, l'UO o le entità dell'account in Organizations, devi disporre di una policy che includa le seguenti operazioni:

- `iam:GenerateOrganizationsAccessReport`
- `iam:GetOrganizationsAccessReport`

- organizations:DescribeAccount
- organizations:DescribeOrganization
- organizations:DescribeOrganizationalUnit
- organizations:DescribePolicy
- organizations:ListChildren
- organizations:ListParents
- organizations:ListPoliciesForTarget
- organizations:ListRoots
- organizations:ListTargetsForPolicy

Per utilizzare l' AWS API AWS CLI or per visualizzare le informazioni sull'ultimo accesso al servizio per Organizations, è necessario disporre di una politica che includa le seguenti azioni:

- iam:GenerateOrganizationsAccessReport
- iam:GetOrganizationsAccessReport
- organizations:DescribePolicy
- organizations:ListChildren
- organizations:ListParents
- organizations:ListPoliciesForTarget
- organizations:ListRoots
- organizations:ListTargetsForPolicy

Questo esempio mostra come creare una policy basata sull'identità che consenta di visualizzare le informazioni sull'ultimo accesso al servizio per Organizations. Inoltre, consente l'accesso in sola lettura a tutto Organizations. Questa policy definisce le autorizzazioni per l'accesso a livello di programmazione e alla console.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GenerateOrganizationsAccessReport",
            "iam:GetOrganizationsAccessReport",
```

```
            "organizations:Describe*",
            "organizations:List*"
        ],
        "Resource": "*"
    }
}
```

Puoi anche utilizzare la chiave iam: OrganizationsPolicyId condition per consentire la generazione di un report solo per una politica specifica di Organizations. Per un esempio di policy, consulta IAM: visualizzazione delle informazioni dell'ultimo accesso al servizio per una policy di Organizations.

Risoluzione dei problemi delle attività per le entità IAM e Organizations

In alcuni casi, l' AWS Management Console ultima tabella delle informazioni a cui si accede potrebbe essere vuota. O forse la tua richiesta AWS CLI o AWS l'API restituisce un set di informazioni vuoto o un campo nullo. In questi casi, verifica i problemi seguenti:

- Per le informazioni sull'ultimo accesso, un'azione che si prevede di visualizzare potrebbe non essere restituita nell'elenco. Ciò può accadere perché l'identità IAM non dispone delle autorizzazioni per l'azione o AWS non tiene ancora traccia dell'azione per le ultime informazioni a cui si accede.
- Per un utente IAM, assicurati che disponga di almeno una policy in linea o gestita collegata, direttamente o tramite le appartenenze ai gruppi.
- Per un gruppo IAM, verifica che disponga di almeno una policy in linea o gestita collegata.
- Per un gruppo IAM, il report restituisce solo i dati sull'ultimo accesso al servizio per i membri che hanno utilizzato le policy del gruppo per accedere a un servizio. Per scoprire se un membro ha utilizzato altre policy, esamina i dati sull'ultimo accesso al servizio per tale utente.
- Per un ruolo IAM, verifica che disponga di almeno una policy in linea o gestita collegata.
- Per un'entità IAM (utente o ruolo), esamina altri tipi di policy che potrebbero influenzare le autorizzazioni di tale entità. Questi includono policy basate sulle risorse, liste di controllo degli accessi, AWS Organizations policy, limiti di autorizzazione IAM o policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy in un singolo account.
- Per una policy IAM, assicurati che la policy gestita specificata sia collegata ad almeno un utente, un gruppo con membri o un ruolo.
- Per un'entità di Organizations (root, UO o account), assicurati di aver effettuato l'accesso utilizzando le credenziali dell'account di gestione di Organizations.
- Verifica che le SCP sono abilitati nel root della tua organizzazione.

- Le informazioni sull'ultimo accesso all'azione sono disponibili solo per alcune azioni elencate in Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM.

Quando apporti modifiche, le attività impiegano almeno quattro ore per comparire nel report della console IAM. Se utilizzi l' AWS API AWS CLI o, devi generare un nuovo rapporto per visualizzare le informazioni aggiornate.

Dove AWS tiene traccia delle ultime informazioni a cui si accede

AWS raccoglie le ultime informazioni a cui si accede per le AWS regioni standard. Quando si AWS aggiungono altre regioni, tali regioni vengono aggiunte alla tabella seguente, inclusa la data di AWS inizio del monitoraggio delle informazioni in ciascuna regione.

- Informazioni sul servizio: il periodo di monitoraggio per i servizi dura almeno 400 giorni, o meno se la regione che ha avviato il monitoraggio di questa funzione negli ultimi 400 giorni.
- Informazioni sulle operazioni: il periodo di monitoraggio delle operazioni di gestione Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio delle operazioni di gestione Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per le operazioni di gestione di tutti gli altri servizi è iniziato il 23 maggio 2023. Se la data di monitoraggio di una regione è successiva al 23 maggio 2023, le informazioni relative all'ultimo accesso all'operazione da quella regione inizieranno da una data successiva.

Nome Regione	Regione	Data di inizio monitoraggio
Stati Uniti orientali (Ohio)	us-east-2	27 ottobre 2017
US East (N. Virginia)	us-east-1	1 Ottobre 2015
US West (N. California)	us-west-1	1 Ottobre 2015
US West (Oregon)	us-west-2	1 Ottobre 2015
Africa (Cape Town)	af-south-1	22 aprile 2020
Asia Pacifico (Hong Kong)	ap-east-1	24 aprile 2019
Asia Pacific (Hyderabad)	ap-south-2	22 novembre 2022

Nome Regione	Regione	Data di inizio monitoraggio
Asia Pacifico (Giacarta)	ap-southeast-3	13 dicembre 2021
Asia Pacifico (Melbourne)	ap-southeast-4	23 gennaio 2023
Asia Pacific (Mumbai)	ap-south-1	27 giugno 2016
Asia Pacifico (Osaka-Locale)	ap-northeast-3	11 febbraio 2018
Asia Pacifico (Seul)	ap-northeast-2	6 gennaio 2016
Asia Pacific (Singapore)	ap-southeast-1	1 Ottobre 2015
Asia Pacific (Sydney)	ap-southeast-2	1 Ottobre 2015
Asia Pacifico (Tokyo)	ap-northeast-1	1 Ottobre 2015
Canada (Central)	ca-central-1	28 ottobre 2017
Europe (Frankfurt)	eu-central-1	1 Ottobre 2015
Europa (Irlanda)	eu-west-1	1 Ottobre 2015
Europe (London)	eu-west-2	28 ottobre 2017
Europa (Milano)	eu-south-1	28 aprile 2020
Europe (Paris)	eu-west-3	18 dicembre 2017
Europa (Spagna)	eu-south-2	15 novembre 2022
Europa (Stoccolma)	eu-north-1	12 dicembre 2018
Europa (Zurigo)	eu-central-2	8 novembre 2022
Israele (Tel Aviv)	il-central-1	1° agosto 2023
Medio Oriente (Bahrein)	me-south-1	29 luglio 2019
Medio Oriente (Emirati Arabi Uniti)	me-central-1	30 agosto 2022

Nome Regione	Regione	Data di inizio monitoraggio
Sud America (São Paulo)	sa-east-1	11 dicembre 2015
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	1 luglio 2023
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	1 luglio 2023

Se una regione non è presente nella tabella precedente, significa che per tale regione non sono ancora disponibili i dati sull'ultimo accesso al servizio.

Una AWS regione è una raccolta di AWS risorse in un'area geografica. Le regioni sono raggruppate in partizioni. Le Regioni standard sono le Regioni che appartengono alla partizione `aws`. Per maggiori informazioni sulle diverse partizioni, consulta il formato Amazon Resource Names (ARN) in Riferimenti generali di AWS. Per ulteriori informazioni sulle regioni, vedere Informazioni sulle AWS regioni anche in Riferimenti generali di AWS.

Visualizzazione delle informazioni sull'ultimo accesso per IAM

Puoi visualizzare le informazioni sull'ultimo accesso per IAM utilizzando l' AWS API AWS Management Console AWS CLI,, o. Consulta un elenco di servizi e delle relative operazioni per i quali vengono visualizzate le informazioni relative all'ultimo accesso. Per ulteriori informazioni sulle ultime informazioni di accesso, vedere Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

È possibile visualizzare le informazioni per i seguenti tipi di risorsa in IAM. In ogni caso, i dati includono i servizi consentiti per il periodo di reporting specificato:

- Utente: visualizza l'ultima volta che l'utente ha tentato di accedere a ogni servizio consentito.
- Gruppo di utenti: visualizza informazioni sull'ultima volta che un membro del gruppo di utenti ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di membri che hanno tentato di accedere.
- Ruolo: visualizza l'ultima volta che qualcuno ha utilizzato il ruolo nel tentativo di accedere a ogni servizio consentito.

- Policy: visualizza le informazioni sull'ultima volta che un utente o un ruolo ha provato ad accedere a ogni servizio consentito. Questo report include anche il numero totale di entità che hanno tentato di accedere.

> **Note**
>
> Prima di visualizzare i dati di accesso per una risorsa in IAM, assicurati di comprendere il periodo di riferimento, le entità incluse nel report e i tipi di policy valutati per i tuoi dati. Per ulteriori dettagli, consulta the section called “Cose da sapere sulle ultime informazioni di accesso”.

Visualizzazione delle informazioni per IAM (console)

È possibile visualizzare le informazioni sull'ultimo accesso per IAM nella scheda Access Advisor della console IAM.

Come visualizzare le informazioni per IAM (console)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Gruppi di utenti, Utenti, Ruoli o Policy.
3. Seleziona il nome di qualsiasi utente, gruppo di utenti, ruolo o policy per aprire la relativa pagina Riepilogo e seleziona la scheda Access Advisor. Visualizzare le seguenti informazioni, in base alla risorsa scelta:

   - Gruppo di utenti: visualizza l'elenco dei servizi a cui i membri del gruppo di utenti possono accedere. È inoltre possibile visualizzare l'ultima volta che un membro ha effettuato l'accesso al servizio, quali policy di gruppo ha utilizzato e quale membro del gruppo ha effettuato la richiesta. Scegli il nome della policy per scoprire se è una policy gestita o una policy del gruppo di utenti in linea. Scegli il nome del membro del gruppo per visualizzare tutti i membri del gruppo di utenti e il momento in cui hanno effettuato l'ultimo accesso al servizio.
   - Utente: visualizza l'elenco dei servizi a cui l'utente può accedere. È inoltre possibile visualizzare l'ultima volta che hanno effettuato l'accesso al servizio e i criteri associati attualmente all’utente. Scegli il nome della policy per sapere se si tratta di una policy gestita, di una policy utente in linea o di una policy in linea per il gruppo di utenti.

- Ruolo: visualizzare l'elenco dei servizi cui il ruolo può accedere, il suo ultimo accesso al servizio e le policy utilizzate. Scegliere il nome della policy per scoprire se è una policy gestita o una policy del ruolo inline.
- Policy: visualizza l'elenco dei servizi con le operazioni consentite nella policy. È inoltre possibile visualizzare l'ultima volta che il criterio è stato utilizzato per accedere al servizio e l'entità (utente o ruolo) utilizzata dal criterio. La data dell'Ultimo accesso include anche quando viene concesso l'accesso a questa policy tramite un'altra policy. Scegliere il nome dell'entità per scoprire a quali entità è collegata questa policy e l'ultimo accesso al servizio da parte dell'entità.

4. Nella colonna Servizio della tabella, scegli il nome di uno dei servizi che include le informazioni relative all'ultimo accesso a un'operazione per visualizzare un elenco delle operazioni di gestione alle quali le entità IAM hanno provato ad accedere. È possibile visualizzare la Regione AWS e un timestamp che indica l'ultimo tentativo di eseguire l'operazione da parte di un utente.
5. La colonna Ultimo accesso viene visualizzata per i servizi e le operazioni di gestione dei servizi che includono le informazioni relative all'ultimo accesso a un'operazione. Esaminare i seguenti risultati possibili restituiti in questa colonna. Questi risultati variano a seconda che un servizio o un'azione sia consentito, sia stato effettuato l'accesso e che venga tracciato AWS per ultimo accesso alle informazioni a cui si accede.

   <number of> giorni fa

   Numero di giorni dall'utilizzo del servizio o dell'azione nel periodo di registrazione. Il periodo di monitoraggio per i servizi è degli ultimi 400 giorni. Il periodo di monitoraggio delle operazioni Amazon S3 è iniziato il 12 aprile 2020. Il periodo di monitoraggio delle azioni di Amazon EC2, IAM e Lambda è iniziato il 7 aprile 2021. Il periodo di monitoraggio per tutti gli altri servizi è iniziato il 23 maggio 2023. Per ulteriori informazioni sulle date di inizio del monitoraggio per ciascuna di esse Regione AWS, consultaDove AWS tiene traccia delle ultime informazioni a cui si accede.

   Non accessibile nel periodo di tracciabilità

   Il servizio o l'azione tracciati non sono stati utilizzati da un'entità nel periodo di registrazione.

   È possibile disporre delle autorizzazioni per un'azione che non viene visualizzata nell'elenco. Ciò può verificarsi se le informazioni di monitoraggio per l'operazione non sono attualmente incluse da AWS. Non è consigliabile prendere decisioni sulle autorizzazioni basate esclusivamente sull'assenza di informazioni di tracciamento. Si consiglia invece di utilizzare queste informazioni

per informare e supportare la strategia generale di concessione di privilegi minimi. Controllare i criteri per verificare che il livello di accesso sia appropriato.

Visualizzazione delle informazioni per IAM (AWS CLI)

Puoi utilizzare il AWS CLI per recuperare informazioni sull'ultima volta che una risorsa IAM è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy.

Come visualizzare le informazioni per IAM (AWS CLI)

1. Generare un report. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. È possibile specificare il livello di granularità che si desidera generare nel report per visualizzare i dettagli di accesso per i servizi o per entrambi i servizi e le azioni. Viene restituito un `job-id` che è possibile utilizzare nelle operazioni `get-service-last-accessed-details` e `get-service-last-accessed-details-with-entities` per monitorare `job-status` finché il processo viene completato.

   - aws iam -details generate-service-last-accessed

2. Recuperare i dettagli sul report utilizzando il parametro `job-id` dal passaggio precedente.

   - aws iam get-service-last-accessed -details

   Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa e il livello di granularità richiesto nell'operazione `generate-service-last-accessed-details`:

   - Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
   - Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Usa l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.
   - Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.

- Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

3. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

   - era iam get-service-last-accessed - details-with-entities

4. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Non restituisce neanche altri tipi di policy, come le policy basate sulle risorse, le liste di controllo accessi, le policy di AWS Organizations , i limiti delle autorizzazioni IAM o le policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy in un singolo account.

   - aws iam list-policies-granting-service -access

Visualizzazione delle informazioni per IAM (AWS API)

Puoi utilizzare l' AWS API per recuperare informazioni sull'ultima volta che una risorsa IAM è stata utilizzata per tentare di accedere ai AWS servizi e alle azioni Amazon S3, Amazon EC2, IAM e Lambda. Una risorsa IAM può essere un utente, un gruppo di utenti, un ruolo o una policy. È possibile specificare il livello di granularità da generare nel report per visualizzare i dettagli relativi ai servizi o ai servizi e alle azioni.

Per visualizzare le informazioni per IAM (API)AWS

1. Generare un report. La richiesta deve includere l'ARN della risorsa IAM (utente, gruppo di utenti, ruolo o policy) per cui desideri un report. Viene restituito un `JobId` che è possibile utilizzare nelle operazioni `GetServiceLastAccessedDetails` e `GetServiceLastAccessedDetailsWithEntities` per monitorare il `JobStatus` finché il processo viene completato.

- GenerateServiceLastAccessedDetails

2. Recuperare i dettagli sul report utilizzando il parametro `JobId` dal passaggio precedente.

   - GetServiceLastAccessedDetails

   Questa operazione restituisce le seguenti informazioni, a seconda del tipo di risorsa e il livello di granularità richiesto nell'operazione `GenerateServiceLastAccessedDetails`:

   - Utente: restituisce un elenco dei servizi cui l'utente specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo dell'utente e l'ARN dell'utente.
   - Gruppo di utenti: restituisce un elenco dei servizi a cui i membri del gruppo di utenti specificato possono accedere utilizzando la policy collegata al gruppo di utenti. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo effettuato da qualsiasi membro del gruppo di utenti. Inoltre, restituisce l'ARN dell'utente e il numero totale di membri del gruppo di utenti che hanno provato ad accedere al servizio. Utilizza l'GetServiceLastAccessedDetailsWithEntitiesoperazione per recuperare un elenco di tutti i membri.
   - Ruolo: restituisce un elenco dei servizi cui il ruolo specificato può accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo del ruolo e l'ARN del ruolo.
   - Policy: restituisce un elenco dei servizi per i quali la policy specificata consente l'accesso. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di accesso al servizio da parte di un'entità (utente o ruolo), utilizzando la policy. Inoltre, restituisce l'ARN di quell'entità e il numero totale di entità che hanno tentato di accedere.

3. Scopri di più sulle entità che hanno utilizzato autorizzazioni di policy o gruppo di utenti in un tentativo di accesso a un servizio specifico. Questa operazione restituisce un elenco delle entità insieme all'ARN, l'ID, il nome, il percorso, il tipo (utente o ruolo) e l'ultimo tentativo di accesso al servizio di ogni entità. È anche possibile utilizzare questa operazione per gli utenti e i ruoli, ma restituisce informazioni solo su tale entità.

   - GetServiceLastAccessedDetailsWithEntities

4. Scopri di più sulle policy basate sull'identità utilizzate da un'identità (utente, gruppo di utenti o ruolo) in un tentativo di accesso a un servizio specifico. Quando si specifica un'identità e un servizio, questa operazione restituisce un elenco delle policy di autorizzazione che l'identità può utilizzare per accedere al servizio specificato. Questa operazione fornisce lo stato attuale delle policy e non dipende dal report generato. Non restituisce neanche altri tipi di policy, come le

policy basate sulle risorse, le liste di controllo accessi, le policy di AWS Organizations , i limiti delle autorizzazioni IAM o le policy di sessione. Per ulteriori informazioni, consulta Tipi di policy o Valutazione delle policy in un singolo account.

- ListPoliciesGrantingServiceAccess

Visualizzazione delle informazioni sull'ultimo accesso per Organizations

Puoi visualizzare le informazioni sull'ultimo accesso al servizio per l' AWS Organizations utilizzo della console IAM o AWS dell'API. AWS CLI Per informazioni importanti sui dati, sulle autorizzazioni necessarie, sulla risoluzione dei problemi e sulle regioni supportate, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Quando accedi alla console IAM utilizzando le credenziali dell'account di AWS Organizations gestione, puoi visualizzare le informazioni relative a qualsiasi entità dell'organizzazione. Le entità di Organizations includono la root dell'organizzazione, le unità organizzative (UO) e gli account. Inoltre, puoi utilizzare la console IAM per visualizzare le informazioni per eventuali policy di controllo dei servizi (SCP) nell'organizzazione. IAM mostra un elenco di servizi consentiti da qualsiasi SCP applicabile all'entità. Per ogni servizio, puoi visualizzare le informazioni sull'attività dell'account più recente per l'entità di Organizations scelta o i suoi elementi figlio.

Quando utilizzi l' AWS API AWS CLI o con le credenziali dell'account di gestione, puoi generare un report per qualsiasi entità o politica dell'organizzazione. Un report programmatico per un'entità include un elenco di servizi consentiti da qualsiasi SCP applicabile all'entità. Per ciascun servizio, il report include l'attività più recente per gli account nell'entità di Organizations specificata o nella sottostruttura dell'entità.

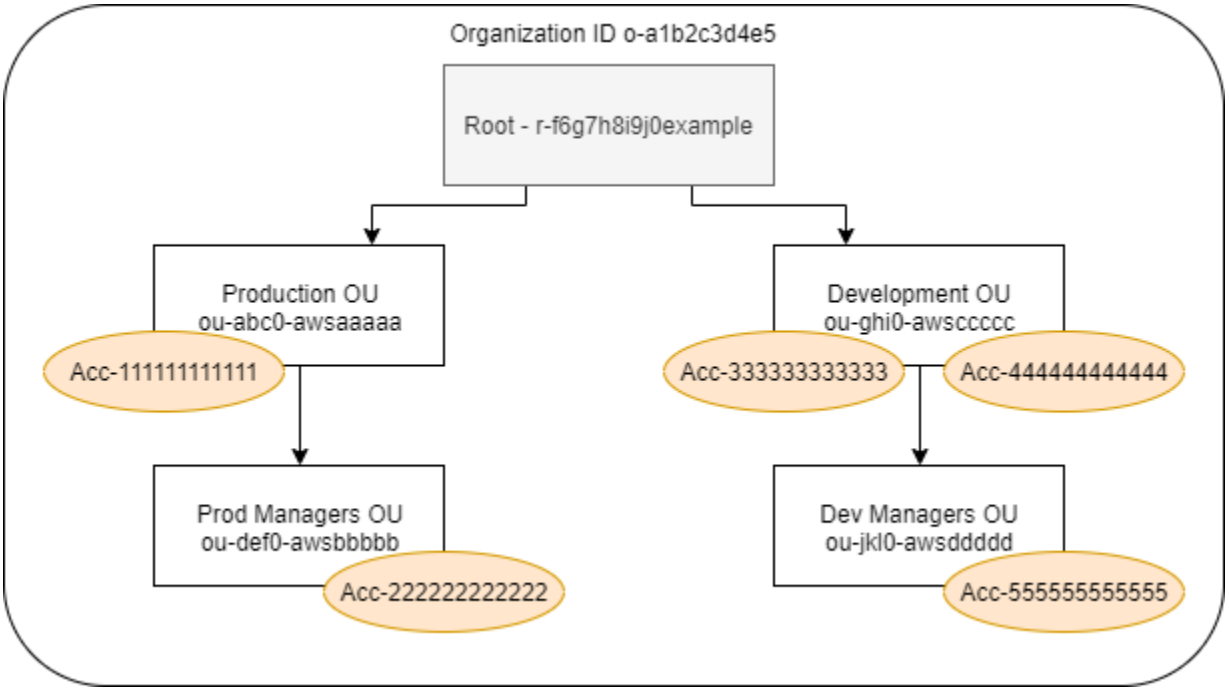
Quando generi un report programmatico per una policy, devi specificare un'entità Organizations. Questo report include un elenco di servizi consentiti dalla SCP specificata. Per ogni servizio, include l'attività dell'account più recente nell'entità o negli elementi figlio dell'entità a cui è concessa l'autorizzazione da tale policy. Per ulteriori informazioni, consulta aws iam generate-organizations-access-report o GenerateOrganizationsAccessReport.

Prima di visualizzare il report, assicurati di aver compreso i requisiti e i dati dell'account di gestione, il periodo del report, le entità incluse nel report e i tipi di policy valutati. Per ulteriori dettagli, consulta the section called “Cose da sapere sulle ultime informazioni di accesso”.

Informazioni sul percorso dell'entità AWS Organizations

Quando si utilizza l' AWS API AWS CLI or per generare un rapporto di AWS Organizations accesso, è necessario specificare il percorso dell'entità. Un percorso è una rappresentazione in testo della struttura di un'entità di Organizations.

È possibile creare un percorso di entità utilizzando la struttura nota dell'organizzazione. Ad esempio, supponiamo di avere la seguente struttura organizzativa AWS Organizations.



Il percorso per l'unità organizzativa Dev Manager viene creato utilizzando gli ID dell'organizzazione, il root e tutte le unità organizzative presenti nel percorso fino all'unità organizzativa inclusa.

```
o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-ghi0-awsccccc/ou-jkl0-awsddddd/
```

Il percorso dell'account nell'unità organizzativa Production (Produzione) viene creato utilizzando gli ID dell'organizzazione, il root, l'unità organizzativa e il numero di account.

```
o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-abc0-awsaaaaa/111111111111/
```

> **Note**
>
> Gli ID organizzazione sono univoci a livello globale, ma gli ID delle unità organizzative e gli ID root sono univoci solo all'interno di un'organizzazione. Ciò significa che non ci sono due organizzazioni che condividono lo stesso ID organizzazione. Tuttavia, un'altra organizzazione

potrebbe avere un'unità organizzativa o un root con il tuo stesso ID. Si consiglia di includere sempre l'ID organizzazione quando si specifica un'unità organizzativa o un root.

Visualizzazione delle informazioni per Organizations (console)

Puoi utilizzare la console IAM per visualizzare le informazioni sull'ultimo accesso al servizio per la root, l'unità organizzativa, l'account o la policy.

Per visualizzare le informazioni relative alla radice (console)

1. Accedi alle credenziali dell'account di gestione AWS Management Console Using Organizations e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.
3. Nella pagina Organization activity (Attività dell'organizzazione), scegliere Root.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). I dati includono un elenco di servizi consentiti dalle policy collegate direttamente alla root. I dati mostrano da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.
5. Scegli la scheda SCP collegate per visualizzare l'elenco di policy di controllo dei servizi (SCP) collegate alla root. IAM mostra il numero di entità di destinazione a cui è collegata ogni policy. È possibile utilizzare queste informazioni per decidere quali SCP rivedere.
6. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
7. Scegli Modifica in AWS Organizations per visualizzare maggiori dettagli e modificare la SCP nella console Organizations. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Per visualizzare le informazioni relative a un'unità organizzativa o a un conto (console)

1. Accedi alle credenziali dell'account di gestione AWS Management Console Using Organizations e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.
3. Nella pagina Organization activity (Attività dell'organizzazione), espandere la struttura dell'organizzazione. Quindi sceglie il nome dell'unità organizzativa o qualsiasi account che si desidera visualizzare, tranne l'account di gestione.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). I dati includono un elenco di servizi consentiti dalle SCP collegate all'unità organizzativa o all'account e a tutti i relativi elementi padre. I dati mostrano da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.
5. Scegli la scheda SCP collegate per visualizzare l'elenco di policy di controllo dei servizi (SCP) collegate direttamente all'unità organizzativa o all'account. IAM mostra il numero di entità di destinazione a cui è collegata ogni policy. È possibile utilizzare queste informazioni per decidere quali SCP rivedere.
6. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
7. Scegli Modifica in AWS Organizations per visualizzare maggiori dettagli e modificare la SCP nella console Organizations. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Come visualizzare le informazioni relative all'account di gestione (console)

1. Accedi alle credenziali dell'account di gestione AWS Management Console Using Organizations e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Attività organizzazione.
3. Nella pagina Attività dell'organizzazione, espandi la struttura dell'organizzazione e scegli il nome dell'account di gestione.
4. Nella scheda Details and activity (Dettagli e attività), visualizzare la sezione Service access report (Report di accesso al servizio). Le informazioni includono un elenco di tutti i servizi AWS . L'account di gestione non è limitato dalle SCP. I dati mostrano se l'account ha effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione. Per maggiori dettagli su quale

principale ha avuto accesso al servizio, accedi come amministratore a tale account e visualizza le informazioni sull'ultimo accesso al servizio IAM.

5. Seleziona la scheda SCP collegate per verificare che non siano presenti SCP collegate, dato che si tratta dell'account di gestione.

Per visualizzare le informazioni relative a un criterio (console)

1. Accedi alle credenziali dell'account di gestione AWS Management Console Using Organizations e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione nella sezione Access reports (Report di accesso), scegliere Policy di controllo dei servizi (SCP).
3. Nella pagina Service control policies (SCPs) (Policy di controllo dei servizi (SCP)), visualizzare un elenco delle policy nell'organizzazione. È possibile visualizzare il numero di entità di destinazione a cui è collegata ogni policy.
4. Scegliere il nome di una SCP per visualizzare tutti i servizi consentiti dalla policy. Per ogni servizio, visualizzare da quale account è stato effettuato l'ultimo accesso al servizio e quando è stata effettuata questa operazione.
5. Scegli Modifica in AWS Organizations per visualizzare maggiori dettagli e modificare la SCP nella console Organizations. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

Visualizzazione delle informazioni per Organizations (AWS CLI)

È possibile utilizzare il AWS CLI per recuperare le informazioni relative all'ultimo accesso al servizio per la radice, l'unità organizzativa, l'account o la policy di Organizations.

Come visualizzare le informazioni sull'ultimo accesso al servizio Organizations (AWS CLI)

1. Utilizza le credenziali dell'account di gestione di Organizations con le autorizzazioni IAM e Organizations richieste e verifica che le SCP siano abilitate per la root. Per ulteriori informazioni, consulta Cose da sapere sulle ultime informazioni di accesso.
2. Generare un report. La richiesta deve includere il percorso dell'entità di Organizations (root, unità organizzativa o account) per cui desideri un report. È anche possibile includere un parametro `organization-policy-id` per visualizzare un report per una policy specifica. Il comando restituisce `job-id` che è quindi possibile utilizzare nel comando `get-organizations-access-report` per monitorare `job-status` fino al completamento del processo.

- era io generate-organizations-access-report

3. Recuperare i dettagli sul report utilizzando il parametro `job-id` dal passaggio precedente.

   - era io get-organizations-access-report

   Questo comando restituisce un elenco di servizi a cui i membri dell'entità possono accedere. Per ogni servizio, il comando restituisce la data e l'ora dell'ultimo tentativo di un membro dell'account e il percorso dell'entità dell'account. Inoltre, restituisce il numero totale di servizi disponibili per l'accesso e il numero di servizi a cui non è stato effettuato l'accesso. Se è stato specificato il parametro `organizations-policy-id` facoltativo, i servizi disponibili per l'accesso sono quelli consentiti dalla policy specificata.

Visualizzazione delle informazioni per Organizations (AWS API)

È possibile utilizzare l' AWS API per recuperare le informazioni relative all'ultimo accesso al servizio per la radice, l'unità organizzativa, l'account o la policy di Organizations.

Per visualizzare le informazioni sull'ultimo accesso (AWS API) al servizio Organizations

1. Utilizza le credenziali dell'account di gestione di Organizations con le autorizzazioni IAM e Organizations richieste e verifica che le SCP siano abilitate per la root. Per ulteriori informazioni, consulta Cose da sapere sulle ultime informazioni di accesso.
2. Generare un report. La richiesta deve includere il percorso dell'entità di Organizations (root, unità organizzativa o account) per cui desideri un report. È anche possibile includere un parametro `OrganizationsPolicyId` per visualizzare un report per una policy specifica. L'operazione restituisce `JobId` che è quindi possibile utilizzare nell'operazione `GetOrganizationsAccessReport` per monitorare `JobStatus` fino al completamento del processo.

   - GenerateOrganizationsAccessReport

3. Recuperare i dettagli sul report utilizzando il parametro `JobId` dal passaggio precedente.

   - GetOrganizationsAccessReport

   Questa operazione restituisce un elenco di servizi a cui i membri dell'entità possono accedere. Per ogni servizio, l'operazione restituisce la data e l'ora dell'ultimo tentativo di un membro

dell'account e il percorso dell'entità dell'account. Inoltre, restituisce il numero totale di servizi disponibili per l'accesso e il numero di servizi a cui non è stato effettuato l'accesso. Se è stato specificato il parametro `OrganizationsPolicyId` facoltativo, i servizi disponibili per l'accesso sono quelli consentiti dalla policy specificata.

Scenari di esempio per l'utilizzo delle ultime informazioni di accesso

Puoi utilizzare le informazioni dell'ultimo accesso per prendere decisioni sulle autorizzazioni da concedere alle tue entità o AWS Organizations entità IAM. Per ulteriori informazioni, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

> **Note**
>
> Prima di visualizzare le informazioni di accesso per un'entità o una policy in IAM AWS Organizations, assicurati di comprendere il periodo di riferimento, le entità segnalate e i tipi di policy valutati per i tuoi dati. Per ulteriori dettagli, consulta the section called “Cose da sapere sulle ultime informazioni di accesso”.

È compito dell'amministratore determinare il giusto equilibrio tra accessibilità e privilegio minimo appropriato per l'azienda.

Utilizzo delle informazioni per ridurre le autorizzazioni per un gruppo IAM

Puoi utilizzare le informazioni sull'ultimo accesso per ridurre le autorizzazioni per un gruppo IAM in modo da includere solo i servizi necessari agli utenti. Questo metodo è una fase importante nella concessione del privilegio minimo a livello di servizio.

Ad esempio, Paulo Santos è l'amministratore responsabile della definizione delle autorizzazioni AWS utente per Example Corp. Questa società ha appena iniziato a utilizzare AWS e il team di sviluppo del software non ha ancora definito AWS i servizi che utilizzerà. Paulo vuole concedere al team l'autorizzazione ad accedere solo ai servizi necessari, ma poiché non sono stati ancora definiti, ha concesso temporaneamente le autorizzazioni di power user. Quindi utilizza le ultime informazioni a cui si accede per ridurre le autorizzazioni del gruppo.

Paulo crea una policy gestita denominata `ExampleDevelopment`, utilizzando il seguente testo in formato JSON. La collega poi a un gruppo denominato `Development` e aggiunge tutti gli sviluppatori al gruppo.

> **Note**
>
> I power user di Paulo potrebbero avere bisogno di autorizzazioni `iam:CreateServiceLinkedRole` per utilizzare alcuni servizi e funzionalità. È consapevole che l'aggiunta di questa autorizzazione consente agli utenti di creare qualsiasi ruolo collegato al servizio. Accetta questo rischio per i suoi power user.

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessToAllServicesExceptPeopleManagement",
            "Effect": "Allow",
            "NotAction": [
                "iam:*",
                "organizations:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequiredIamAndOrgsActions",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

Paulo decide di attendere 90 giorni prima di visualizzare i dati sull'ultimo accesso al servizio per il gruppo `Development` utilizzando la AWS Management Console. Visualizza l'elenco dei servizi a cui i membri del gruppo hanno effettuato l'accesso. Viene a sapere che gli utenti hanno avuto accesso a cinque servizi nell'ultima settimana: AWS CloudTrail Amazon CloudWatch Logs, Amazon EC2 AWS KMS e Amazon S3. Hanno avuto accesso ad alcuni altri servizi durante la prima valutazione AWS, ma non da allora.

Paulo decide di ridurre le autorizzazioni della policy in modo da includere solo cinque servizi e le operazioni IAM e Organizations necessarie. Modifica la policy `ExampleDevelopment` utilizzando il seguente testo in formato JSON.

> **Note**
>
> I power user di Paulo potrebbero avere bisogno di autorizzazioni `iam:CreateServiceLinkedRole` per utilizzare alcuni servizi e funzionalità. È consapevole che l'aggiunta di questa autorizzazione consente agli utenti di creare qualsiasi ruolo collegato al servizio. Accetta questo rischio per i suoi power user.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccessToListedServices",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "kms:*",
                "cloudtrail:*",
                "logs:*",
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequiredIamAndOrgsActions",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

Per ridurre ulteriormente le autorizzazioni, Paulo può visualizzare gli eventi dell'account nella AWS CloudTrail Cronologia degli eventi. Qui può visualizzare informazioni dettagliate sugli eventi, che può utilizzare per ridurre le autorizzazioni della policy in modo da includere solo le operazioni e le risorse di cui hanno bisogno gli sviluppatori. Per ulteriori informazioni, consulta Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Utilizzo delle informazioni per ridurre le autorizzazioni per un utente IAM

Puoi utilizzare le informazioni sull'ultimo accesso per ridurre le autorizzazioni per un singolo utente IAM.

Ad esempio, Martha Rivera è un'amministratrice IT responsabile di garantire che i dipendenti della sua azienda non dispongano di autorizzazioni eccessive AWS . Come parte di un controllo periodico di sicurezza, l'amministratore esamina le autorizzazioni di tutti gli utenti IAM. Uno di questi utenti, Nikhil Jayashankar, è uno sviluppatore di applicazioni, che in precedenza ha ricoperto il ruolo di tecnico della sicurezza. A causa del cambiamento dei requisiti della mansione, Nikhil è membro sia del gruppo `app-dev` sia del gruppo `security-team`. Il gruppo `app-dev` per la sua nuova mansione concede autorizzazioni per più servizi, tra cui Amazon EC2, Amazon EBS, Auto Scaling, Amazon S3, Route 53 e Elastic Transcoder. Il `security-team` gruppo per il suo vecchio lavoro concede le autorizzazioni a IAM e. CloudTrail

L'amministratore Martha accede alla console IAM e seleziona Utenti, quindi sceglie il nome `nikhilj` e sceglie la scheda Access Advisor.

Martha esamina la colonna Ultimo accesso e nota che Nikhil non ha recentemente effettuato l'accesso a IAM, Route 53 CloudTrail, Amazon Elastic Transcoder e a diversi altri servizi. AWS Nikhil ha effettuato l'accesso ad Amazon S3. Martha sceglie S3 dall'elenco dei servizi e apprende che Nikhil ha eseguito alcune azioni `List` S3 nelle ultime due settimane. All'interno della sua azienda, Martha conferma che Nikhil non ha alcuna necessità aziendale di accedere a IAM e che non CloudTrail fa più parte del team di sicurezza interno.

Martha è ora pronta ad agire in base al servizio e all'azione dell'ultimo accesso alle informazioni. Tuttavia, diversamente da quanto avviene con il gruppo dell'esempio precedente, un utente IAM come `nikhilj` potrebbe essere soggetto a più policy ed essere membro di più gruppi. Martha deve procedere con cautela per evitare di interrompere inavvertitamente l'accesso per `nikhilj` o per altri membri del gruppo. Oltre a conoscere il tipo di accesso di cui Nikhil deve disporre, deve determinare il modo in cui Nikhil riceve le autorizzazioni.

Martha sceglie la scheda Permissions (Autorizzazioni), dove visualizza le policy collegate direttamente a `nikhilj` e quelle collegate da un gruppo. Espande ogni policy e visualizza il riepilogo per scoprire quale policy consente l'accesso ai servizi che Nikhil non sta utilizzando:

- IAM: la policy `IAMFullAccess` AWS gestita è allegata direttamente `nikhilj` e allegata al gruppo. `security-team`
- CloudTrail — La policy `AWS CloudTrailReadOnlyAccess` AWS gestita è allegata al `security-team` gruppo.
- Route 53: la policy gestita dal cliente `App-Dev-Route53` è collegata al gruppo `app-dev`.
- Elastic Transcoder: la policy gestita dal cliente `App-Dev-ElasticTranscoder` è collegata al gruppo `app-dev`.

Martha decide di rimuovere la policy `IAMFullAccess` AWS gestita a cui è allegata direttamente. `nikhilj` Rimuove inoltre l'appartenenza di Nikhil al gruppo `security-team`. Queste due azioni rimuovono l'accesso non necessario a IAM e CloudTrail.

Le autorizzazioni di Nikhil per accedere a Route 53 ed Elastic Transcoder sono concesse dal gruppo `app-dev`. Anche se Nikhil non li utilizza, questi servizi potrebbero essere utili ad altri membri del gruppo. Martha esamina le informazioni sull'ultimo accesso per il gruppo `app-dev` e scopre che diversi membri hanno recentemente effettuato l'accesso a Route 53 e Amazon S3. Ma nessun membro del gruppo ha avuto accesso a Elastic Transcoder nell'ultimo anno. Rimuove quindi dal gruppo la policy gestita dal cliente `App-Dev-ElasticTranscoder`.

Martha esamina poi i dati sull'ultimo accesso al servizio per la policy gestita dal cliente `App-Dev-ElasticTranscoder`. Scopre che la policy non è collegata a nessun'altra identità IAM. Indaga all'interno della sua azienda per accertarsi che la policy non sarà necessaria in futuro e quindi la elimina.

Utilizzo delle informazioni prima dell'eliminazione delle risorse IAM

Puoi utilizzare le informazioni sull'ultimo accesso al servizio prima di eliminare una risorsa IAM per assicurarti che sia trascorso un determinato intervallo di tempo dall'ultimo utilizzo di tale risorsa. Questo si applica a utenti, gruppi, ruoli e policy. Per ulteriori informazioni su queste operazioni, consulta i seguenti argomenti:

- Utenti: Eliminazione di un utente
- Gruppi: Eliminazione di un gruppo
- Ruoli: Eliminazione di un ruolo

- Policy: Eliminazione di una policy gestita (comporta inoltre lo scollegamento della policy dalle identità)

Utilizzo delle informazioni prima della modifica delle policy IAM

Puoi esaminare le informazioni sull'ultimo accesso per un'identità IAM (utente, gruppo o ruolo) o per una policy IAM prima di modificare una policy che influisce sulla risorsa. È un'opzione importante per non rimuovere l'accesso per qualcuno che la utilizza.

Ad esempio, Arnav Desai è sviluppatore e AWS amministratore di Example Corp. Quando il suo team ha iniziato a utilizzare AWS, ha fornito a tutti gli sviluppatori un accesso da utente esperto che consentiva loro l'accesso completo a tutti i servizi tranne IAM e Organizations. Come primo passo verso la concessione del privilegio minimo, Arnav desidera utilizzare il per rivedere le politiche gestite nel suo account. AWS CLI

A tale scopo, Arnav elenca innanzitutto le policy di autorizzazione gestite dal cliente nel suo account che sono collegate a un'identità, utilizzando il seguente comando:

```
aws iam list-policies --scope Local --only-attached --policy-usage-filter
 PermissionsPolicy
```

Dalla risposta, acquisisce l'ARN per ogni policy. Arnav genera quindi un report relativo ai dati sull'ultimo accesso al servizio per ogni policy, utilizzando il comando seguente.

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Da questa risposta, acquisisce l'ID del report generato dal campo `JobId`. Arnav testa il comando seguente finché il campo `JobStatus` restituisce un valore `COMPLETED` o `FAILED`. Se il processo ha esito negativo, acquisisce l'errore.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Quando lo stato del processo diventa `COMPLETED`, Arnav analizza i contenuti dell'array `ServicesLastAccessed` in formato JSON.

```
 "ServicesLastAccessed": [
        {
            "TotalAuthenticatedEntities": 1,
```

```
            "LastAuthenticated": 2018-11-01T21:24:33.222Z,
            "ServiceNamespace": "dynamodb",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/IAMExampleUser",
            "ServiceName": "Amazon DynamoDB"
        },

        {
            "TotalAuthenticatedEntities": 0,
            "ServiceNamespace": "ec2",
            "ServiceName": "Amazon EC2"
        },

        {
            "TotalAuthenticatedEntities": 3,
            "LastAuthenticated": 2018-08-25T15:29:51.156Z,
            "ServiceNamespace": "s3",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:role/IAMExampleRole",
            "ServiceName": "Amazon S3"
        }
    ]
```

Da queste informazioni, Arnav apprende che la policy `ExamplePolicy1` consente l'accesso a tre servizi: Amazon DynamoDB, Amazon S3 e Amazon EC2. L'utente IAM denominato `IAMExampleUser` ha provato accedere l'ultima volta a DynamoDB il 1° novembre e qualcuno ha utilizzato il ruolo `IAMExampleRole` per provare ad accedere ad Amazon S3 il 25 agosto. Ci sono anche altre due entità che hanno provato ad accedere ad Amazon S3 nell'ultimo anno. Tuttavia, nessuno ha provato ad accedere ad Amazon EC2 nell'ultimo anno.

Questo significa che Arnav può rimuovere in modo sicuro le operazioni Amazon EC2 dalla policy. Arnav vuole esaminare l'attuale documento JSON per la policy. In primo luogo, deve determinare il numero di versione della policy utilizzando il comando seguente.

```
aws iam list-policy-versions --policy-arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Dalla risposta, Arnav raccoglie l'attuale numero di versione predefinita dall'array `Versions`. Utilizza quindi quel numero di versione (`v2`) per richiedere il documento JSON della policy utilizzando il comando seguente.

```
aws iam get-policy-version --policy-arn arn:aws:iam::123456789012:policy/ExamplePolicy1
 --version-id v2
```

Arnav memorizza il documento JSON della policy restituito nel campo `Document` dell'array `PolicyVersion`. Nel documento della policy, Arnav ricerca le operazioni nello spazio dei nomi `ec2`. Se non ci sono operazioni dagli altri spazi dei nomi rimanenti nella policy, scollega la policy dalle identità interessate (utenti, gruppi e ruoli) e la elimina. In questo caso, la policy include i servizi Amazon DynamoDB e Amazon S3. Pertanto, Arnav rimuove le operazioni Amazon EC2 dal documento e salva le modifiche. Utilizza quindi il seguente comando per aggiornare la policy con la nuova versione del documento e impostare tale versione come versione predefinita della policy.

```
aws iam create-policy-version --policy-arn arn:aws:iam::123456789012:policy/
ExamplePolicy1 --policy-document file://UpdatedPolicy.json --set-as-default
```

La policy `ExamplePolicy1` è ora aggiornata per rimuovere l'accesso al servizio Amazon EC2 non necessario.

Altri scenari IAM

Le informazioni sull'ultimo tentativo di accesso di una risorsa IAM (utente, gruppo, ruolo o policy) a un servizio possono essere utili per completare una delle seguenti attività:

- Policy: Modifica di una policy esistente in linea o gestita dal cliente per rimuovere le autorizzazioni
- Policy: Conversione di una policy in linea in una policy gestita e successiva eliminazione
- Policy: Aggiunta di un rifiuto esplicito a una policy esistente
- Policy: Scollegamento di una policy gestita da un'identità (utente, gruppo o ruolo)
- Entità: Impostazione di un limite di autorizzazioni per controllare il numero massimo di autorizzazioni che un'entità (utente o ruolo) può avere
- Gruppi: Rimozione di utenti da un gruppo

Utilizzo dei dati per perfezionare le autorizzazioni di un'unità organizzativa

Puoi utilizzare i dati sull'ultimo accesso al servizio per perfezionare le autorizzazioni per un'unità organizzativa in AWS Organizations.

Ad esempio, John Stiles è un amministratore. AWS Organizations È responsabile di garantire che le persone in azienda Account AWS non dispongano di autorizzazioni eccessive. Come parte di un audit di sicurezza periodico, esamina le autorizzazioni dell'organizzazione. La sua `Development` unità organizzativa contiene account che vengono spesso utilizzati per testare nuovi AWS servizi. John decide di controllare periodicamente il report per servizi a cui non è stato effettuato alcun

accesso da più di 180 giorni. Quindi, rimuove le autorizzazioni concesse ai membri dell'unità organizzativa per accedere a questi servizi.

John esegue l'accesso alla console IAM utilizzando le sue credenziali dell'account di gestione. Nella console IAM, individua i dati di Organizations per l'unità organizzativa `Development`. Esamina la tabella dei rapporti sull'accesso ai servizi e vede due AWS servizi a cui non si accede da più di 180 giorni dal suo periodo preferito di 180 giorni. Ricorda di aver aggiunto le autorizzazioni per i team di sviluppo per accedere ad Amazon Lex e. AWS Database Migration Service John contatta i team di sviluppo e conferma che non hanno più l'esigenza aziendale di testare questi servizi.

John è pronto a usare le ultime informazioni di accesso. Sceglie Modifica in AWS Organizations e gli viene segnalato che la SCP è collegata a più entità. Sceglie Continue (Continua). Nel AWS Organizations, esamina gli obiettivi per sapere a quali entità Organizations è collegato l'SCP. Tutte le entità si trovano all'interno dell'unità organizzativa `Development`.

John decide di negare l'accesso ad Amazon Lex e alle AWS Database Migration Service azioni in `NewServiceTest` SCP. Questa operazione rimuove l'accesso non necessario ai servizi.

Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM

La tabella seguente elenca i AWS servizi per i quali vengono visualizzate le informazioni sull'ultimo accesso all'azione IAM. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

Servizio	Prefisso del servizio
AWS Identity and Access Management Access Analyzer	access-analyzer
AWS Account Management	account
AWS Certificate Manager	acm
Flussi di lavoro gestiti da Amazon per Apache Airflow	airflow
AWS Amplify	amplify
AWS Amplify Generatore di interfacce utente	amplifyuibuilder
Amazon AppIntegrations	app-integrations

Servizio	Prefisso del servizio
AWS AppConfig	appconfig
Amazon AppFlow	appflow
AWS Application Cost Profiler	application-cost-profiler
Informazioni approfondite sulle CloudWatch applicazioni Amazon	applicationinsights
AWS App Mesh	appmesh
Amazon AppStream 2.0	appstream
AWS AppSync	appsync
Amazon Managed Service per Prometheus	aps
Amazon Athena	athena
AWS Audit Manager	auditmanager
AWS Auto Scaling	autoscaling
Marketplace AWS	aws-marketplace
AWS Backup	backup
AWS Batch	batch
Amazon Braket	braket
Budget AWS	budgets
AWS Cloud9	cloud9
AWS CloudFormation	cloudformation
Amazon CloudFront	cloudfront

Servizio	Prefisso del servizio
AWS CloudHSM	cloudhsm
Amazon CloudSearch	cloudsearch
AWS CloudTrail	cloudtrail
Amazon CloudWatch	cloudwatch
AWS CodeArtifact	codeartifact
AWS CodeDeploy	codedeploy
Amazon CodeGuru Profiler	codeguru-profiler
CodeGuru Revisore Amazon	codeguru-reviewer
AWS CodePipeline	codepipeline
AWS CodeStar	codestar
Notifiche AWS CodeStar	codestar-notifications
Amazon Cognito Identity	cognito-identity
Pool di utenti Amazon Cognito	cognito-idp
Amazon Cognito Sync	cognito-sync
Amazon Comprehend Medical	comprehendmedical
AWS Compute Optimizer	compute-optimizer
AWS Config	config
Amazon Connect	connect
AWS Cost and Usage Report	cur

Servizio	Prefisso del servizio
AWS Glue DataBrew	databrew
AWS Data Exchange	dataexchange
AWS Data Pipeline	datapipeline
DynamoDB Accelerator	dax
AWS Device Farm	devicefarm
Amazon DevOps Guru	devops-guru
AWS Direct Connect	directconnect
Amazon Data Lifecycle Manager	dlm
AWS Database Migration Service	dms
Cluster elastici Amazon DocumentDB	docdb-elastic
AWS Directory Service	ds
Amazon DynamoDB	dynamodb
Amazon Elastic Block Store	ebs
Amazon Elastic Compute Cloud	ec2
Amazon Elastic Container Registry	ecr
Amazon Elastic Container Registry Public	ecr-public
Amazon Elastic Container Service	ecs
Amazon Elastic Kubernetes Service	eks
Amazon Elastic Inference	elastic-inference
Amazon ElastiCache	elasticache

Servizio	Prefisso del servizio
AWS Elastic Beanstalk	elasticbeanstalk
Amazon Elastic File System	elasticfilesystem
Elastic Load Balancing	elasticloadbalancing
Amazon Elastic Transcoder	elastictranscoder
Amazon EMR su EKS (Containers EMR)	emr-containers
Amazon EMR Serverless	emr-serverless
OpenSearch Servizio Amazon	es
Amazon EventBridge	events
Amazon CloudWatch evidentemente	evidently
Amazon FinSpace	finspace
Amazon Data Firehose	firehose
AWS Fault Injection Service	fis
AWS Firewall Manager	fms
Amazon Fraud Detector	frauddetector
Amazon FSx	fsx
Amazon GameLift	gamelift
Servizio di posizione Amazon	geo
Amazon S3 Glacier	glacier
Grafana gestito da Amazon	grafana

Servizio	Prefisso del servizio
AWS IoT Greengrass	greengrass
AWS Ground Station	groundstation
Amazon GuardDuty	guardduty
AWS HealthLake	healthlake
Amazon Honeycode	honeycode
AWS Identity and Access Management	iam
AWS Archivio di identità	identitystore
EC2 Image Builder	imagebuilder
Amazon Inspector Classic	inspector
Amazon Inspector	inspector2
AWS IoT	iot
AWS IoT Analytics	iotanalytics
AWS IoT Core Device Advisor	iotdeviceadvisor
AWS IoT Events	iotevents
AWS IoT Fleet Hub	iotfleethub
AWS IoT SiteWise	iotsitewise
AWS IoT TwinMaker	iottwinmaker
AWS IoT Wireless	iotwireless
Amazon Interactive Video Service	ivs
Amazon Interactive Video Service Chat	ivschat

Servizio	Prefisso del servizio
Amazon Managed Streaming per Apache Kafka	kafka
Amazon Managed Streaming per Kafka Connect	kafkaconnect
Amazon Kendra	kendra
Amazon Kinesis	kinesis
Amazon Kinesis Analytics V2	kinesisanalytics
AWS Key Management Service	kms
AWS Lambda	lambda
Amazon Lex	lex
AWS License Manager Gestore di abbonamenti Linux	license-manager-linux-subscriptions
Amazon Lightsail	lightsail
CloudWatch Registri Amazon	log
Amazon Lookout per le apparecchiature	lookoutequipment
Amazon Lookout per le metriche	lookoutmetrics
Amazon Lookout per Vision	lookoutvision
AWS Mainframe Modernization	m2
Blockchain gestita da Amazon	managedblockchain
AWS Elemental MediaConnect	mediaconnect
AWS Elemental MediaConvert	mediaconvert
AWS Elemental MediaLive	medialive

Servizio	Prefisso del servizio
AWS Elemental MediaStore	mediastore
AWS Elemental MediaTailor	mediatailor
Amazon MemoryDB per Redis	memorydb
AWS Application Migration Service	mgn
AWS Migration Hub	mgh
AWS Suggerimenti di strategia dell'Hub di migrazione	migration hub-strategy
Amazon Pinpoint	mobiletargeting
Amazon MQ	mq
AWS Network Manager	networkmanager
Amazon Nimble Studio	nimble
AWS HealthOmics	omics
AWS OpsWorks	opsworks
AWS OpsWorks CM	opsworks-cm
AWS Outposts	outposts
AWS Organizations	organizations
AWS Panorama	panorama
AWS Performance Insights	pi
EventBridgeTubi Amazon	pipes
Amazon Polly	polly

Servizio	Prefisso del servizio
Profili cliente Amazon Connect	profilo
Amazon QLDB	qldb
AWS Resource Access Manager	ram
AWS Cestino di riciclaggio	rbin
Amazon Relational Database Service	rds
Amazon Redshift	redshift
API dati di Amazon Redshift	redshift-data
AWS Migration Hub Refactor Spaces	refactor-spaces
Amazon Rekognition	rekognition
AWS Resilience Hub	resiliencehub
Esploratore di risorse AWS	resource-explorer-2
AWS Resource Groups	resource-groups
AWS RoboMaker	robomaker
AWS Identity and Access Management Ruoli ovunque	rolesanywhere
Amazon Route 53	route53
Controlli di ripristino Amazon Route 53	route53-recovery-control-config
Preparazione al ripristino di Amazon Route 53	route53-recovery-readiness
Amazon Route 53 Resolver	route53resolver

Servizio	Prefisso del servizio
AWS CloudWatchRUM	rum
Amazon Simple Storage Service	s3
Amazon S3 su Outposts	s3-outposts
Funzionalità SageMaker geospaziali di Amazon	sagemaker-geospatial
Savings Plans	savingsplans
EventBridgeSchemi Amazon	schemas
Amazon SimpleDB	sdb
AWS Secrets Manager	secretsmanager
AWS Security Hub	securityhub
Amazon Security Lake	securitylake
AWS Serverless Application Repository	serverlessrepo
AWS Service Catalog	servicecatalog
AWS Cloud Map	servicediscovery
Service Quotas	servicequotas
Amazon Simple Email Service	ses
AWS Shield	shield
AWS Signer	signer
AWS SimSpace Weaver	simspaceweaver
AWS Server Migration Service	sms

Servizio	Prefisso del servizio
Servizio di SMS e messaggi vocali Amazon Pinpoint	sms-voice
AWS Snowball	snowball
Amazon Simple Queue Service	sqs
AWS Systems Manager	ssm
AWS Systems Manager Incident Manager	ssm-incidents
AWS Systems Manager per SAP	ssm-sap
AWS Step Functions	states
AWS Security Token Service	sts
Amazon Simple Workflow Service	swf
Amazon CloudWatch Synthetics	synthetics
AWS Resource Groups Tagging API	tag
Amazon Textract	textract
Amazon Timestream	timestream
AWS Costruttore di reti di telecomunicazioni	tnb
Amazon Transcribe	transcribe
AWS Transfer Family	transfer
Amazon Translate	translate
Amazon Connect Voice ID	voiceid
Amazon VPC Lattice	vpc-lattice
AWS WAFV2	wafv2

Servizio	Prefisso del servizio
AWS Well-Architected Tool	wellarchitected
Amazon Connect Wisdom	wisdom
Amazon WorkLink	worklink
Amazon WorkSpaces	workspace
AWS X-Ray	xray

Operazioni per le quali sono disponibili le informazioni relative all'ultimo accesso

La tabella seguente elenca le operazioni per le quali sono disponibili le informazioni relative all'ultimo accesso all'operazione stessa.

Prefisso del servizio	Azioni
access-analyzer	access-analyzer: Regola ApplyArchive
	access-analyzer: generazione CancelPolicy
	analizzatore di accesso: CheckAccess NotGranted
	analizzatore di accesso: CheckNo NewAccess
	access-analyzer: Anteprima CreateAccess
	analizzatore di accesso: CreateAnalyzer
	access-analyzer: Regola CreateArchive
	analizzatore di accesso: DeleteAnalyzer
	access-analyzer: Regola DeleteArchive
	access-analyzer: Anteprima GetAccess
	access-analyzer: Risorsa GetAnalyzed

Prefisso del servizio	Azioni
	analizzatore di accesso: GetAnalyzer
	access-analyzer: Regola GetArchive
	analizzatore di accesso: GetFinding
	access-analyzer: politica GetGenerated
	analizzatore di accesso: ListAccess PreviewFindings
	access-analyzer: anteprime ListAccess
	access-analyzer: ListAnalyzed Risorse
	analizzatore di accesso: ListAnalyzers
	access-analyzer: Regole ListArchive
	analizzatore di accesso: ListFindings
	access-analyzer: generazioni ListPolicy
	access-analyzer: Generazione StartPolicy
	access-analyzer: Scan StartResource
	access-analyzer: Regola UpdateArchive
	analizzatore di accesso: UpdateFindings
	analizzatore di accesso: ValidatePolicy

Prefisso del servizio	Azioni
account	account: contatto DeleteAlternate conto: DisableRegion conto: EnableRegion account: GetAlternate Contatti account: GetContact Informazioni conto: GetRegion OptStatus conto: ListRegions account: PutAlternate Contatti account: PutContact Informazioni
acm	cam: DeleteCertificate acm: DescribeCertificate acm: ExportCertificate acm: Configurazione GetAccount acm: GetCertificate acm: ImportCertificate acm: ListCertificates acm: Configurazione PutAccount acm: RenewCertificate acm: RequestCertificate acm: Posta elettronica ResendValidation acm: Opzioni UpdateCertificate

Prefisso del servizio	Azioni
airflow	flusso d'aria: Token CreateCli flusso d'aria: CreateEnvironment flusso d'aria: CreateWeb LoginToken flusso d'aria: DeleteEnvironment flusso d'aria: GetEnvironment flusso d'aria: ListEnvironments flusso d'aria: PublishMetrics flusso d'aria: UpdateEnvironment

Prefisso del servizio	Azioni
amplify	amplificare: CreateApp
	amplify: Ambiente CreateBackend
	amplificare: CreateBranch
	amplificare: CreateDeployment
	amplify: Associazione CreateDomain
	amplify: Hook CreateWeb
	amplificare: DeleteApp
	amplify: Ambiente DeleteBackend
	amplificare: DeleteBranch
	amplify: Associazione DeleteDomain
	amplify: DeleteJob
	amplificare: gancio DeleteWeb
	amplify: Registri GenerateAccess
	amplificare: GetApp
	amplify: Url GetArtifact
	amplify: Ambiente GetBackend
	amplificare: GetBranch
	amplify: Associazione GetDomain
	amplify: GetJob
	amplificare: gancio GetWeb
	amplificare: ListApps

Prefisso del servizio	Azioni
	amplificare: ListArtifacts amplify: Ambienti ListBackend amplificare: ListBranches amplify: Associazioni ListDomain amplificare: ListJobs amplificare: ganci ListWeb amplificare: StartDeployment amplificare: StartJob amplificare: StopJob amplificare: UpdateApp amplificare: UpdateBranch amplify: Associazione UpdateDomain amplify: Hook UpdateWeb

Prefisso del servizio	Azioni
amplifyuibuilder	amplifyuibuilder: CreateComponent amplificyuibuilder: CreateForm amplificyuibuilder: CreateTheme amplificyuibuilder: DeleteComponent amplificyuibuilder: DeleteForm amplificyuibuilder: DeleteTheme amplificyuibuilder: ExportComponents amplificyuibuilder: ExportThemes amplifyuibuilder: Job GetCodegen amplifyuibuilder: Offerte di lavoro ListCodegen amplifyuibuilder: ListComponents amplificyuibuilder: ListForms amplificyuibuilder: ListThemes amplifyuibuilder: Bandiera ResetMetadata amplifyuibuilder: Job StartCodegen amplifyuibuilder: UpdateComponent amplificyuibuilder: UpdateForm amplificyuibuilder: UpdateTheme

Prefisso del servizio	Azioni
app-integrations	integrazioni con app: CreateApplication
	integrazioni con app: integrazione CreateData
	integrazioni di app: integrazione CreateEvent
	integrazioni con app: DeleteApplication
	integrazioni con app: integrazione DeleteData
	integrazioni di app: integrazione DeleteEvent
	integrazioni con app: GetApplication
	integrazioni con app: integrazione GetData
	integrazioni di app: integrazione GetEvent
	integrazioni di app: associazioni ListApplication
	integrazioni con app: ListApplications
	integrazioni con app: ListData IntegrationAssociations
	integrazioni di app: integrazioni ListData
	integrazioni di app: ListEvent IntegrationAssociations
	integrazioni di app: integrazioni ListEvent
	integrazioni di app: UpdateApplication
	integrazioni con app: integrazione UpdateData
	integrazioni di app: integrazione UpdateEvent

Prefisso del servizio	Azioni
appconfig	appconfig: CreateApplication appconfig: Profilo CreateConfiguration appconfig: Strategia CreateDeployment appconfig: CreateEnvironment app config: CreateExtension appconfig: Associazione CreateExtension appconfig: CreateHosted ConfigurationVersion app config: DeleteApplication appconfig: Profilo DeleteConfiguration appconfig: Strategia DeleteDeployment appconfig: DeleteEnvironment app config: DeleteExtension appconfig: Associazione DeleteExtension appconfig: DeleteHosted ConfigurationVersion app config: GetApplication app config: GetConfiguration appconfig: Profilo GetConfiguration appconfig: GetDeployment appconfig: Strategia GetDeployment appconfig: GetEnvironment app config: GetExtension

Prefisso del servizio	Azioni
	appconfig: Associazione GetExtension appconfig: GetHosted ConfigurationVersion app config: ListApplications appconfig: Profili ListConfiguration appconfig: ListDeployments appconfig: strategie ListDeployment appconfig: ListEnvironments appconfig: Associazioni ListExtension appconfig: ListExtensions app config: ListHosted ConfigurationVersions app config: StartDeployment app config: StopDeployment app config: UpdateApplication appconfig: Profilo UpdateConfiguration appconfig: Strategia UpdateDeployment appconfig: UpdateEnvironment app config: UpdateExtension appconfig: Associazione UpdateExtension appconfig: ValidateConfiguration

Prefisso del servizio	Azioni
appflow	appflow: esecuzioni CancelFlow
	appflow: Profilo CreateConnector
	appflow: CreateFlow
	appflow: Profilo DeleteConnector
	appflow: DeleteFlow
	flusso di app: DescribeConnector
	appflow: Entità DescribeConnector
	appflow: Profili DescribeConnector
	appflow: DescribeConnectors
	flusso di app: DescribeFlow
	flusso di app: DescribeFlow ExecutionRecords
	appflow: Entità ListConnector
	appflow: ListConnectors
	flusso di app: ListFlows
	flusso di app: RegisterConnector
	flusso di app: ResetConnector MetadataCache
	flusso di app: StartFlow
	flusso di app: StopFlow
	appflow: connettore UnRegister
	appflow: Profilo UpdateConnector
	appflow: Registrazione UpdateConnector

Prefisso del servizio	Azioni
	appflow: UpdateFlow
application-cost-profiler	application-cost-profiler: definizione DeleteReport application-cost-profiler: definizione GetReport application-cost-profiler: utilizzo ImportApplication application-cost-profiler: definizioni ListReport application-cost-profiler: definizione PutReport application-cost-profiler: definizione UpdateReport

Prefisso del servizio	Azioni
applicationinsights	approfondimenti sulle applicazioni: AddWorkload
	approfondimenti sulle applicazioni: CreateApplication
	approfondimenti sulle applicazioni: CreateComponent
	approfondimenti sulle applicazioni: modello CreateLog
	approfondimenti sulle applicazioni: DeleteApplication
	approfondimenti sulle applicazioni: DeleteComponent
	approfondimenti sulle applicazioni: modello DeleteLog
	approfondimenti sulle applicazioni: DescribeApplication
	approfondimenti sulle applicazioni: DescribeComponent
	approfondimenti sulle applicazioni: configurazione DescribeComponent
	approfondimenti sulle applicazioni: DescribeComponent ConfigurationRecommendation
	approfondimenti sulle applicazioni: modello DescribeLog
	approfondimenti sulle applicazioni: DescribeObservation
	approfondimenti sulle applicazioni: DescribeProblem
	approfondimenti sulle applicazioni: osservazioni DescribeProblem
	approfondimenti sulle applicazioni: DescribeWorkload
	approfondimenti sulle applicazioni: ListApplications
	approfondimenti sulle applicazioni: ListComponents
	applicationinsights: Storia ListConfiguration
	approfondimenti sulle applicazioni: modelli ListLog

Prefisso del servizio	Azioni
	approfondimenti sulle applicazioni: ListLog PatternSets
	approfondimenti sulle applicazioni: ListProblems
	approfondimenti sulle applicazioni: ListWorkloads
	approfondimenti sulle applicazioni: RemoveWorkload
	approfondimenti sulle applicazioni: UpdateApplication
	approfondimenti sulle applicazioni: UpdateComponent
	approfondimenti sulle applicazioni: configurazione UpdateCom ponent
	approfondimenti sulle applicazioni: modello UpdateLog
	approfondimenti sulle applicazioni: UpdateWorkload

Prefisso del servizio	Azioni
appmesh	appmesh: Percorso CreateGateway appmesh: CreateMesh app mesh: CreateRoute appmesh: gateway CreateVirtual appmesh: Nodo CreateVirtual appmesh: router CreateVirtual appmesh: Servizio CreateVirtual appmesh: Percorso DeleteGateway appmesh: DeleteMesh app mesh: DeleteRoute appmesh: gateway DeleteVirtual appmesh: Nodo DeleteVirtual appmesh: router DeleteVirtual appmesh: Servizio DeleteVirtual appmesh: Percorso DescribeGateway appmesh: DescribeMesh app mesh: DescribeRoute appmesh: gateway DescribeVirtual appmesh: Nodo DescribeVirtual appmesh: router DescribeVirtual appmesh: Servizio DescribeVirtual

Prefisso del servizio	Azioni
	appmesh: percorsi ListGateway appmesh: ListMeshes app mesh: ListRoutes appmesh: gateway ListVirtual appmesh: nodi ListVirtual appmesh: router ListVirtual appmesh: Servizi ListVirtual appmesh: Risorse StreamAggregated appmesh: Percorso UpdateGateway appmesh: UpdateMesh app mesh: UpdateRoute appmesh: gateway UpdateVirtual appmesh: Nodo UpdateVirtual appmesh: router UpdateVirtual appmesh: Servizio UpdateVirtual

Prefisso del servizio	Azioni
appstream	appstream: AssociateApp BlockBuilder AppBlock
	appstream: Flotta AssociateApplication
	appstream: AssociateApplication ToEntitlement
	appstream: AssociateFleet
	appstream: BatchAssociate UserStack
	appstream: BatchDisassociate UserStack
	appstream: CopyImage
	appstream: Blocca CreateApp
	appstream: CreateApp BlockBuilder
	appstream: URL di streaming CreateApp BlockBuilder
	appstream: CreateApplication
	appstream: Config CreateDirectory
	appstream: CreateEntitlement
	appstream: CreateFleet
	appstream: Generatore CreateImage
	appstream: URL CreateImage BuilderStreaming
	appstream: CreateStack
	appstream: URL CreateStreaming
	appstream: Immagine CreateUpdated
	appstream: CreateUsage ReportSubscription
	appstream: CreateUser

Prefisso del servizio	Azioni
	appstream: Blocca DeleteApp
	appstream: DeleteApp BlockBuilder
	appstream: DeleteApplication
	appstream: Config DeleteDirectory
	appstream: DeleteEntitlement
	appstream: DeleteFleet
	appstream: DeleteImage
	appstream: Generatore DeleteImage
	appstream: Autorizzazioni DeleteImage
	appstream: DeleteStack
	appstream: DeleteUsage ReportSubscription
	appstream: DeleteUser
	appstream: Associazioni DescribeApp BlockBuilder AppBlock
	appstream: DescribeApp BlockBuilders
	appstream: Blocchi DescribeApp
	appstream: DescribeApplication FleetAssociations
	appstream: DescribeApplications
	appstream: configurazioni DescribeDirectory
	appstream: DescribeEntitlements
	appstream: DescribeFleets
	appstream: Costruttori DescribeImage

Prefisso del servizio	Azioni
	appstream: Autorizzazioni DescribeImage
	appstream: DescribeImages
	appstream: DescribeSessions
	appstream: DescribeStacks
	appstream: DescribeUsage ReportSubscriptions
	appstream: DescribeUsers
	appstream: DescribeUser StackAssociations
	appstream: DisableUser
	appstream: DisassociateApp BlockBuilder AppBlock
	appstream: Flotta DisassociateApplication
	appstream: DisassociateApplication FromEntitlement
	appstream: DisassociateFleet
	appstream: EnableUser
	appstream: ExpireSession
	appstream: flotte ListAssociated
	appstream: pile ListAssociated
	appstream: Applicazioni ListEntitled
	appstream: StartApp BlockBuilder
	appstream: StartFleet
	appstream: Generatore StartImage
	appstream: StopApp BlockBuilder

Prefisso del servizio	Azioni
	appstream: StopFleet
	appstream: Generatore StopImage
	appstream: UpdateApp BlockBuilder
	appstream: UpdateApplication
	appstream: Config UpdateDirectory
	appstream: UpdateEntitlement
	appstream: UpdateFleet
	appstream: Autorizzazioni UpdateImage
	appstream: UpdateStack

Prefisso del servizio	Azioni
appsync	sincronizzazione delle app: AssociateApi sincronizzazione delle app: AssociateMerged GraphqlApi sincronizzazione delle app: AssociateSource GraphqlApi appsync: cache CreateApi appsync: chiave CreateApi appsync: fonte CreateData appsync: nome CreateDomain appsync: CreateFunction appsync: API CreateGraphql sincronizzazione app: CreateResolver sincronizzazione app: CreateType appsync: cache DeleteApi appsync: chiave DeleteApi appsync: fonte DeleteData appsync: nome DeleteDomain appsync: DeleteFunction appsync: API DeleteGraphql sincronizzazione app: DeleteResolver sincronizzazione app: DeleteType sincronizzazione app: DisassociateApi sincronizzazione delle app: DisassociateMerged GraphqlApi

Prefisso del servizio	Azioni
	sincronizzazione delle app: DisassociateSource GraphqlApi sincronizzazione delle app: EvaluateCode appsync: modello EvaluateMapping appsync: cache FlushApi appsync: Associazione GetApi appsync: cache GetApi appsync: fonte GetData appsync: GetData SourceIntrospection appsync: nome GetDomain appsync: GetFunction appsync: API GetGraphql appsync: variabili GetGraphql ApiEnvironment appsync: schema GetIntrospection sincronizzazione dell'app: GetResolver sincronizzazione delle app: GetSchema CreationStatus sincronizzazione delle app: GetSource ApiAssociation sincronizzazione delle app: GetType appsync: chiavi ListApi appsync: Sorgenti ListData appsync: nomi ListDomain appsync: ListFunctions

Prefisso del servizio	Azioni
	appsync: API ListGraphql sincronizzazione dell'app: ListResolvers sincronizzazione delle app: ListResolvers ByFunction sincronizzazione delle app: ListSource ApiAssociations sincronizzazione delle app: ListTypes sincronizzazione delle app: ListTypes ByAssociation appsync: variabili PutGraphql ApiEnvironment appsync: StartData SourceIntrospection appsync: creazione StartSchema appsync: Unisci StartSchema appsync: cache UpdateApi appsync: chiave UpdateApi appsync: fonte UpdateData appsync: nome UpdateDomain appsync: UpdateFunction appsync: API UpdateGraphql sincronizzazione app: UpdateResolver sincronizzazione delle app: UpdateSource ApiAssociation sincronizzazione delle app: UpdateType

Prefisso del servizio	Azioni
aps	app: CreateAlert ManagerDefinition aps: CreateLogging Configurazione Mappe: CreateRule GroupsNamespace rubinetti: CreateScraper rubinetti: CreateWorkspace rubinetti: DeleteAlert ManagerDefinition aps: DeleteLogging Configurazione Mappe: DeleteRule GroupsNamespace rubinetti: DeleteScraper rubinetti: DeleteWorkspace rubinetti: DescribeAlert ManagerDefinition aps: DescribeLogging Configurazione Mappe: DescribeRule GroupsNamespace rubinetti: DescribeScraper rubinetti: DescribeWorkspace rubinetti: GetDefault ScraperConfiguration rubinetti: ListRule GroupsNamespaces rubinetti: ListScrapers rubinetti: ListWorkspaces rubinetti: PutAlert ManagerDefinition rubinetti: PutRule GroupsNamespace

Prefisso del servizio	Azioni
	aps: UpdateLogging Configurazione aps: UpdateWorkspace alias

Prefisso del servizio	Azioni
athena	atena: BatchGet NamedQuery atena: BatchGet PreparedStatement atena: BatchGet QueryExecution athena: Prenotazione CancelCapacity athena: Prenotazione CreateCapacity athena: Catalogo CreateData athena: Domanda CreateNamed atena: CreateNotebook athena: Dichiarazione CreatePrepared athena: CreatePresigned NotebookUrl athena: Gruppo CreateWork athena: Prenotazione DeleteCapacity athena: Catalogo DeleteData athena: Domanda DeleteNamed atena: DeleteNotebook athena: Dichiarazione DeletePrepared athena: Gruppo DeleteWork athena: ExportNotebook athena: Esecuzione GetCalculation athena: GetCalculation ExecutionCode atena: GetCalculation ExecutionStatus

Prefisso del servizio	Azioni
	atena: GetCapacity AssignmentConfiguration athena: Prenotazione GetCapacity athena: GetDatabase athena: Catalogo GetData athena: Domanda GetNamed athena: Metadati GetNotebook athena: Dichiarazione GetPrepared athena: Esecuzione GetQuery athena: Risultati GetQuery athena: GetQuery ResultsStream atena: GetQuery RuntimeStatistics atena: GetSession athena: stato GetSession athena: Metadati GetTable athena: Gruppo GetWork athena: ImportNotebook athena: DPUSizes ListApplication athena: Esecuzioni ListCalculation athena: Prenotazioni ListCapacity athena: ListDatabases athena: Cataloghi ListData

Prefisso del servizio	Azioni
	athena: Versioni ListEngine athena: ListExecutors athena: Domande ListNamed athena: Metadati ListNotebook athena: Sessioni ListNotebook athena: Dichiarazioni ListPrepared athena: Esecuzioni ListQuery atena: ListSessions athena: Metadati ListTable athena: Gruppi ListWork athena: PutCapacity AssignmentConfiguration athena: Esecuzione StartCalculation athena: Esecuzione StartQuery athena: StartSession athena: Esecuzione StopCalculation athena: Esecuzione StopQuery athena: TerminateSession athena: Prenotazione UpdateCapacity athena: Catalogo UpdateData athena: Domanda UpdateNamed atena: UpdateNotebook

Prefisso del servizio	Azioni
	athena: Metadati UpdateNotebook athena: Dichiarazione UpdatePrepared athena: Gruppo UpdateWork

Prefisso del servizio	Azioni
auditmanager	auditmanager: Cartella AssociateAssessment ReportEvidence
	auditmanager: Evidenza BatchAssociate AssessmentReport
	auditmanager: Valutazione BatchCreate DelegationBy
	auditmanager: Valutazione BatchDelete DelegationBy
	auditmanager: Evidenza BatchDisassociate AssessmentReport
	responsabile dell'audit: BatchImport EvidenceTo Assessmen tControl
	responsabile dell'audit: CreateAssessment
	auditmanager: struttura CreateAssessment
	auditmanager: rapporto CreateAssessment
	responsabile dell'audit: CreateControl
	responsabile dell'audit: DeleteAssessment
	auditmanager: struttura DeleteAssessment
	gestore di controllo: DeleteAssessment FrameworkShare
	auditmanager: rapporto DeleteAssessment
	responsabile dell'audit: DeleteControl
	responsabile dell'audit: DeregisterAccount
	responsabile dell'audit: DeregisterOrganization AdminAccount
	auditmanager: Cartella DisassociateAssessment ReportEvidence
	auditmanager: Stato GetAccount
	responsabile di controllo: GetAssessment

Prefisso del servizio	Azioni
	auditmanager: struttura GetAssessment
	gestore di controllo: GetAssessment ReportUrl
	auditmanager: registri GetChange
	gestore di controllo: GetControl
	responsabile dell'audit: GetDelegations
	responsabile dell'audit: GetEvidence
	auditmanager: Cartella GetEvidence ByEvidence
	auditmanager: Url GetEvidence FileUpload
	auditmanager: Cartella GetEvidence
	auditmanager: Valutazione GetEvidence FoldersBy
	responsabile dell'audit: GetEvidence FoldersBy AssessmentControl
	responsabile dell'audit: GetInsights
	responsabile dell'audit: GetInsights ByAssessment
	responsabile dell'audit: GetOrganization AdminAccount
	responsabile dell'audit: GetServices InScope
	responsabile dell'audit: GetSettings
	auditmanager: Dominio ListAssessment ControlInsights ByControl
	auditmanager: Framework ListAssessment
	auditmanager: Richieste ListAssessment FrameworkShare
	auditmanager: Rapporti ListAssessment
	responsabile di controllo: ListAssessments

Prefisso del servizio	Azioni
	responsabile dell'audit: ListControl DomainInsights
	responsabile dell'audit: ListControl DomainInsights ByAssessment
	responsabile dell'audit: ListControl InsightsBy ControlDomain
	responsabile dell'audit: ListControls
	auditmanager: fonte ListKeywords ForData
	gestore di controllo: ListNotifications
	responsabile dell'audit: RegisterAccount
	responsabile dell'audit: RegisterOrganization AdminAccount
	responsabile dell'audit: StartAssessment FrameworkShare
	responsabile dell'audit: UpdateAssessment
	auditmanager: Controllo UpdateAssessment
	auditmanager: Stato UpdateAssessment ControlSet
	auditmanager: Framework UpdateAssessment
	gestore di controllo: UpdateAssessment FrameworkShare
	auditmanager: Stato UpdateAssessment
	responsabile di controllo: UpdateControl
	responsabile dell'audit: UpdateSettings
	responsabile dell'audit: ValidateAssessment ReportIntegrity

Prefisso del servizio	Azioni
scalabilità automatica	scalabilità automatica: AttachInstances scalabilità automatica: bilanciatori AttachLoad scalabilità automatica AttachLoadBalancerTarget: gruppi scalabilità automatica: sorgenti AttachTraffic scalabilità automatica: BatchDelete ScheduledAction scalabilità automatica: BatchPut ScheduledUpdate GroupAction scalabilità automatica: aggiorna CancelInstance scalabilità automaticaCompleteLifecycle: azione scalabilità automatica: CreateAuto ScalingGroup scalabilità automatica: configurazione CreateLaunch scalabilità automatica: DeleteAuto ScalingGroup scalabilità automatica: configurazione DeleteLaunch scalabilità automatica: Hook DeleteLifecycle scalabilità automatica: configurazione DeleteNotification scalabilità automatica: DeletePolicy scalabilità automatica: azione DeleteScheduled scalabilità automatica: Pool DeleteWarm scalabilità automatica: limiti DescribeAccount scalabilità automatica: tipi DescribeAdjustment scalabilità automatica: DescribeAuto ScalingGroups scalabilità automatica: DescribeAuto ScalingInstances

Prefisso del servizio	Azioni
	scalabilità automatica: tipi DescribeAuto ScalingNotification scalabilità automatica: aggiorna DescribeInstance scalabilità automatica: DescribeLaunch configurazioni scalabilità automatica: DescribeLifecycle Hooks scalabilità automatica: DescribeLifecycle HookTypes scalabilità automatica: bilanciatori DescribeLoad scalabilità automatica DescribeLoadBalancerTarget: gruppi scalabilità automatica: DescribeMetric CollectionTypes scalabilità automatica: configurazioni DescribeNotification scalabilità automatica: DescribePolicies scalabilità automatica: attività DescribeScaling scalabilità automatica: DescribeScaling ProcessTypes scalabilità automatica: azioni DescribeScheduled scalabilità automatica: DescribeTermination PolicyTypes scalabilità automatica: sorgenti DescribeTraffic scalabilità automatica: Pool DescribeWarm scalabilità automatica: DetachInstances scalabilità automatica: bilanciatori DetachLoad scalabilità automatica DetachLoadBalancerTarget: gruppi scalabilità automatica: sorgenti DetachTraffic autoscaling: raccolta DisableMetrics

Prefisso del servizio	Azioni
	autoscaling: Collezione EnableMetrics scalabilità automatica: EnterStandby scalabilità automatica: ExecutePolicy scalabilità automatica: ExitStandby scalabilità automatica: GetPredictive ScalingForecast scalabilità automatica: Hook PutLifecycle scalabilità automatica: configurazione PutNotification scalabilità automatica: politica PutScaling scalabilità automatica: azione PutScheduled UpdateGroup scalabilità automatica: Pool PutWarm scalabilità automatica: RecordLifecycle ActionHeartbeat scalabilità automatica: ResumeProcesses scalabilità automatica: aggiorna RollbackInstance scalabilità automaticaSetDesired: capacità scalabilità automatica: Health SetInstance scalabilità automatica: protezione SetInstance scalabilità automatica: aggiornamento StartInstance scalabilità automatica: SuspendProcesses scalabilità automatica: TerminateInstance InAuto ScalingGroup scalabilità automatica: UpdateAuto ScalingGroup
aws-marketplace	aws-marketplace: GetEntitlements

Prefisso del servizio	Azioni
backup	backup: tieni premuto CancelLegal backup: CreateBackup Pianifica backup: CreateBackup selezione backup: CreateBackup Vault backup: CreateFramework backup: CreateLegal tieni premuto backup: CreateLogically AirGapped BackupVault backup: CreateReport Pianifica backup: CreateRestore TestingPlan backup: CreateRestore TestingSelection backup: DeleteBackup Pianifica backup: DeleteBackup selezione backup: DeleteBackup Vault backup: politica DeleteBackup VaultAccess backup: DeleteBackup VaultLock configurazione backup: DeleteBackup VaultNotifications backup: DeleteFramework backup: DeleteRecovery Punto backup: DeleteReport Pianifica backup: DeleteRestore TestingPlan backup: DeleteRestore TestingSelection

Prefisso del servizio	Azioni
	backup: DescribeBackup Job backup: DescribeBackup Vault backup: DescribeCopy Job backup: DescribeFramework backup: DescribeGlobal impostazioni backup: DescribeProtected risorsa backup: DescribeRecovery Punto backup: DescribeRegion impostazioni backup: DescribeReport Job backup: DescribeReport Pianifica backup: DescribeRestore Job backup: DisassociateRecovery Punto backup: DisassociateRecovery PointFrom Genitore backup: ExportBackup PlanTemplate backup: GetBackup Pianifica backup: GetBackup PlanFrom JSON backup: modello GetBackup PlanFrom backup: GetBackup selezione backup: GetBackup VaultAccess politica backup: GetBackup VaultNotifications backup: GetLegal tieni premuto

Prefisso del servizio	Azioni
	backup: GetRecovery PointRestore metadati backup: GetRestore JobMetadata backup: GetRestore TestingInferred metadati backup: GetRestore TestingPlan backup: GetRestore TestingSelection backup: GetSupported ResourceTypes backup: ListBackup Offerte di lavoro backup: ListBackup JobSummaries backup: ListBackup piani backup: ListBackup PlanTemplates backup: ListBackup PlanVersions backup: ListBackup selezioni backup: ListBackup Vaults backup: Offerte di lavoro ListCopy backup: ListCopy JobSummaries backup: ListFrameworks backup: ListLegal contiene backup: ListProtected risorse backup: ListRecovery PointsBy BackupVault backup: ListRecovery PointsBy LegalHold backup: ListRecovery PointsBy risorsa

Prefisso del servizio	Azioni
	backup: ListReport Offerte di lavoro backup: ListReport Piani backup: ListRestore Offerte di lavoro backup: ListRestore JobsBy ProtectedResource backup: ListRestore JobSummaries backup: ListRestore TestingPlans backup: ListRestore TestingSelections backup: PutBackup VaultAccess politica backup: PutBackup VaultLock configurazione backup: PutBackup VaultNotifications backup: PutRestore ValidationResult backup: StartBackup Job backup: StartCopy Job backup: StartReport Job backup: StartRestore Job backup: StopBackup Job backup: UpdateBackup Pianifica backup: UpdateFramework backup: UpdateGlobal impostazioni backup: UpdateRecovery PointLifecycle backup: UpdateRegion impostazioni

Prefisso del servizio	Azioni
	backup: UpdateReport Pianifica backup: UpdateRestore TestingPlan backup: UpdateRestore TestingSelection

Prefisso del servizio	Azioni
batch	lotto: CancelJob batch: CreateCompute Ambiente batch: CreateJob Coda batch: politica CreateScheduling batch: DeleteCompute Ambiente batch: DeleteJob Coda batch: politica DeleteScheduling batch: DeregisterJob definizione batch: DescribeCompute Ambienti batch: DescribeJob definizioni batch: DescribeJob code lotto: DescribeJobs batch: DescribeScheduling politiche lotto: ListJobs batch: ListScheduling politiche batch: RegisterJob definizione lotto: SubmitJob lotto: TerminateJob batch: UpdateCompute Ambiente batch: UpdateJob Coda batch: politica UpdateScheduling

Prefisso del servizio	Azioni
braket	parentesi: accordo AcceptUser staffa: caratteristica AccessBraket staffa: CancelJob staffa: Task CancelQuantum staffa: CreateJob staffa: Task CreateQuantum staffa: GetDevice staffa: GetJob staffa: Task GetQuantum staffa: Status GetService LinkedRole staffa: GetUser AgreementStatus staffa: SearchDevices staffa: SearchJobs staffa: compiti SearchQuantum

Prefisso del servizio	Azioni
budgets	bilanci: ModifyBudget budget: azione CreateBudget bilanci: ModifyBudget bilanci: ModifyBudget bilanci: ModifyBudget budget: azione DeleteBudget bilanci: ModifyBudget bilanci: ModifyBudget bilanci: ViewBudget budget: azione DescribeBudget budget: DescribeBudget ActionHistories budget: conto DescribeBudget ActionsFor budget: Budget DescribeBudget ActionsFor budget: ViewBudget bilanci: ViewBudget bilanci: ViewBudget bilanci: ViewBudget bilanci: ViewBudget bilanci: azione ExecuteBudget bilanci: ModifyBudget bilanci: azione UpdateBudget

Prefisso del servizio	Azioni
	bilanci: ModifyBudget bilanci: ModifyBudget
cloud9	cloud9: EC2 CreateEnvironment cloud9: iscrizione CreateEnvironment cloud9: DeleteEnvironment cloud9: iscrizione DeleteEnvironment cloud9: Abbonamenti DescribeEnvironment cloud 9: DescribeEnvironments cloud9: stato DescribeEnvironment nuvola 9: ListEnvironments nuvola 9: UpdateEnvironment cloud9: iscrizione UpdateEnvironment

Prefisso del servizio	Azioni
cloudformation	formazione del cloud: BatchDescribe TypeConfigurations formazione nel cloud: Stack CancelUpdate cloudformation: rollback ContinueUpdate cloudformation: Imposta CreateChange cloudformation: modello CreateGenerated formazione di nuvole: CreateStack cloudformation: Istanze CreateStack cloudformation: Imposta CreateStack formazione di nuvole: DeactivateType cloudformation: impostato DeleteChange cloudformation: modello DeleteGenerated formazione di nuvole: DeleteStack cloudformation: Istanze DeleteStack cloudformation: Imposta DeleteStack formazione di nuvole: DeregisterType cloudformation: limiti DescribeAccount cloudformation: Imposta DescribeChange formazione di nuvole: DescribeChange SetHooks cloudformation: modello DescribeGenerated cloudformation: Accesso DescribeOrganizations formazione del cloud: DescribePublisher

Prefisso del servizio	Azioni
	cloudformation: scansione DescribeResource
	cloudformation: stato DescribeStack DriftDetection
	cloudformation: Eventi DescribeStack
	cloudformation: Istanza DescribeStack
	cloudformation: risorsa DescribeStack
	formazione di nuvole: DescribeStack ResourceDrifts
	cloudformation: risorse DescribeStack
	formazione del cloud: DescribeStacks
	cloudformation: impostato DescribeStack
	formazione di nuvole: DescribeStack SetOperation
	formazione di nuvole: DescribeType
	cloudformation: registrazione DescribeType
	cloudformation: Drift DetectStack
	formazione di nuvole: DetectStack ResourceDrift
	formazione di nuvole: DetectStack SetDrift
	cloudformation: costo EstimateTemplate
	cloudformation: impostato ExecuteChange
	cloudformation: modello GetGenerated
	cloudformation: politica GetStack
	formazione di nuvole: GetTemplate
	cloudformation: riepilogo GetTemplate

Prefisso del servizio	Azioni
	cloudformation: Imposta ImportStacks ToStack
	cloudformation: set ListChange
	formazione di nuvole: ListExports
	cloudformation: modelli ListGenerated
	formazione di nuvole: ListImports
	cloudformation: risorse ListResource ScanRelated
	formazione del cloud: ListResource ScanResources
	cloudformation: scansioni ListResource
	cloudformation: derive ListStack InstanceResource
	cloudformationListStack: Istanze
	cloudformation: risorse ListStack
	formazione del cloud: ListStack SetAuto DeploymentTargets
	cloudformation: Risultati ListStack SetOperation
	formazione di nuvole: ListStack SetOperations
	cloudformation: set ListStack
	cloudformation: registrazioni ListType
	formazione di nuvole: ListTypes
	cloudformation: versioni ListType
	formazione di nuvole: PublishType
	cloudformation: progresso RecordHandler
	formazione di nuvole: RegisterPublisher

Prefisso del servizio	Azioni
	formazione di nuvole: RegisterType
	formazione di nuvole: RollbackStack
	cloudformation: politica SetStack
	cloudformation: configurazione SetType
	formazione di nuvole: SetType DefaultVersion
	formazione di nuvole: SignalResource
	cloudformation: scansione StartResource
	formazione di nuvole: StopStack SetOperation
	formazione di nuvole: TestType
	cloudformation: modello UpdateGenerated
	formazione di nuvole: UpdateStack
	cloudformation: Istanze UpdateStack
	cloudformation: Imposta UpdateStack
	cloudformation: protezione UpdateTermination
	formazione di nuvole: ValidateTemplate

Prefisso del servizio	Azioni
cloudfront	fronte cloud: AssociateAlias cloudfront: politica CreateCache cloudfront: CreateCloud FrontOrigin AccessIdentity fronte cloud: CreateContinuous DeploymentPolicy cloudfront: Config CreateField LevelEncryption cloudfront: Profilo CreateField LevelEncryption cloudfront: CreateFunction fronte cloud: CreateInvalidation cloudfront: Gruppo CreateKey cloudfront: CreateKey ValueStore cloudfront: abbonamento CreateMonitoring cloudfront: CreateOrigin AccessControl fronte cloud: CreateOrigin RequestPolicy cloudfront: chiave CreatePublic cloudfront: CreateRealtime LogConfig fronte cloud: CreateResponse HeadersPolicy cloudfront: politica DeleteCache cloudfront: DeleteCloud FrontOrigin AccessIdentity fronte cloud: DeleteContinuous DeploymentPolicy fronte cloud: DeleteDistribution cloudfront: Config DeleteField LevelEncryption

Prefisso del servizio	Azioni
	cloudfront: Profilo DeleteField LevelEncryption cloudfront: DeleteFunction cloudfront: Gruppo DeleteKey cloudfront: DeleteKey ValueStore cloudfront: abbonamento DeleteMonitoring cloudfront: DeleteOrigin AccessControl fronte cloud: DeleteOrigin RequestPolicy cloudfront: chiave DeletePublic cloudfront: DeleteRealtime LogConfig fronte cloud: DeleteResponse HeadersPolicy cloudfront: distribuzione DeleteStreaming cloudfront: DescribeFunction fronte cloud: DescribeKey ValueStore cloudfront: politica GetCache cloudfront: GetCache PolicyConfig fronte cloud: GetCloud FrontOrigin AccessIdentity cloudfront: Config GetCloud FrontOrigin AccessIdentity cloudfront: GetContinuous DeploymentPolicy cloudfront: Config GetContinuous DeploymentPolicy cloudfront: Config GetDistribution cloudfront: GetField LevelEncryption

Prefisso del servizio	Azioni
	cloudfront: Config GetField LevelEncryption cloudfront: Profilo GetField LevelEncryption cloudfront: GetField LevelEncryption ProfileConfig fronte cloud: GetFunction fronte cloud: GetInvalidation cloudfront: Gruppo GetKey cloudfront: GetKey GroupConfig cloudfront: abbonamento GetMonitoring cloudfront: GetOrigin AccessControl cloudfront: Config GetOrigin AccessControl cloudfront: GetOrigin RequestPolicy cloudfront: Config GetOrigin RequestPolicy cloudfront: chiave GetPublic cloudfront: GetPublic KeyConfig fronte cloud: GetRealtime LogConfig fronte cloud: GetResponse HeadersPolicy cloudfront: Config GetResponse HeadersPolicy cloudfront: Distribuzione GetStreaming cloudfront: GetStreaming DistributionConfig cloudfront: politiche ListCache cloudfront: ListCloud FrontOrigin AccessIdentities

Prefisso del servizio	Azioni
	cloudfront: alias ListConflicting cloudfront: ListContinuous DeploymentPolicies fronte cloud: ListDistributions fronte cloud: ListDistributions ByCache PolicyId cloudfront: Gruppo ListDistributions ByKey cloudfront: ID ListDistributions ByOrigin RequestPolicy fronte cloud: ListDistributions ByRealtime LogConfig cloudfront: ID ListDistributions ByResponse HeadersPolicy cloudfronte:acLID ListDistributions ByWeb cloudfront: configurazioni ListField LevelEncryption cloudfront: Profili ListField LevelEncryption cloudfront: ListFunctions fronte cloud: ListInvalidations cloudfront: gruppi ListKey cloudfront: ListKey ValueStores fronte cloud: ListOrigin AccessControls fronte cloud: ListOrigin RequestPolicies cloudfront: chiavi ListPublic cloudfront: ListRealtime LogConfigs fronte cloud: ListResponse HeadersPolicies cloudfront: distribuzioni ListStreaming

Prefisso del servizio	Azioni
	cloudfront: PublishFunction fronte cloud: TestFunction cloudfront: politica UpdateCache cloudfront: UpdateCloud FrontOrigin AccessIdentity fronte cloud: UpdateContinuous DeploymentPolicy fronte cloud: UpdateDistribution cloudfront: Config UpdateField LevelEncryption cloudfront: Profilo UpdateField LevelEncryption cloudfront: UpdateFunction cloudfront: Gruppo UpdateKey cloudfront: UpdateKey ValueStore fronte cloud: UpdateOrigin AccessControl fronte cloud: UpdateOrigin RequestPolicy cloudfront: chiave UpdatePublic cloudfront: UpdateRealtime LogConfig fronte cloud: UpdateResponse HeadersPolicy

Prefisso del servizio	Azioni
cloudhsm	cloudhsm: CreateHapg
	cloudhsm: CreateHsm
	cloudhsm: Cliente CreateLuna
	cloudhsm: DeleteBackup
	cloudhsm: DeleteHapg
	cloudhsm: DeleteHsm
	cloudhsm: Cliente DeleteLuna
	cloudhsm: DescribeBackups
	cloudhsm: DescribeClusters
	cloudhsm: DescribeHapg
	cloudhsm: DescribeHsm
	cloudhsm: Cliente DescribeLuna
	cloudhsm: GetConfig
	cloudhsm: InitializeCluster
	cloudhsm: zone ListAvailable
	cloudhsm: ListHapgs
	cloudhsm: ListHsms
	cloudhsm: Clienti ListLuna
	cloudhsm: Attributi ModifyBackup
	cloudhsm: ModifyCluster
	cloudhsm: ModifyHapg

Prefisso del servizio	Azioni
	cloudhsm: ModifyHsm cloudhsm: Cliente ModifyLuna cloudhsm: RestoreBackup

Prefisso del servizio	Azioni
cloudsearch	ricerca nel cloud: BuildSuggesters ricerca nel cloud: CreateDomain cloudsearch: schema DefineAnalysis ricerca nel cloud: DefineExpression cloudsearch: Campo DefineIndex ricerca nel cloud: DefineSuggester cloudsearch: schema DeleteAnalysis ricerca nel cloud: DeleteDomain ricerca nel cloud: DeleteExpression cloudsearch: Campo DeleteIndex ricerca nel cloud: DeleteSuggester cloudsearch: schemi DescribeAnalysis cloudsearch: Opzioni DescribeAvailability cloudsearch: DescribeDomain EndpointOptions ricerca nel cloud: DescribeDomains ricerca nel cloud: DescribeExpressions cloudsearch: Campi DescribeIndex cloudsearch: Parametri DescribeScaling cloudsearch: DescribeService AccessPolicies ricerca nel cloud: DescribeSuggesters ricerca nel cloud: IndexDocuments

Prefisso del servizio	Azioni
	cloudsearch: nomi ListDomain cloudsearch: Opzioni UpdateAvailability cloudsearch: UpdateDomain EndpointOptions cloudsearch: parametri UpdateScaling cloudsearch: UpdateService AccessPolicies

Prefisso del servizio	Azioni
cloudtrail	pista nuvolosa: CancelQuery
	pista nuvolosa: CreateChannel
	pista nuvolosa: CreateEvent DataStore
	pista nuvolosa: CreateTrail
	pista nuvolosa: DeleteChannel
	pista nuvolosa: DeleteEvent DataStore
	cloudtrail: politica DeleteResource
	cloudtrail: DeleteTrail
	pista nuvolosa: DeregisterOrganization DelegatedAdmin
	pista nuvolosa: DescribeQuery
	pista nuvolosa: DescribeTrails
	pista nuvolosa: DisableFederation
	pista nuvolosa: GetChannel
	pista nuvolosa: GetEvent DataStore
	cloudtrail: dati GetEvent DataStore
	cloudtrail: selettori GetEvent
	cloudtrail: GetImport
	cloudtrail: selettori GetInsight
	cloudtrail: Risultati GetQuery
	cloudtrail: politica GetResource
	cloudtrail: GetTrail

Prefisso del servizio	Azioni
	cloudtrail: stato GetTrail
	cloudtrail: ListChannels
	pista nuvolosa: ListEvent DataStores
	cloudtrail: errori ListImport
	cloudtrail: ListImports
	cloudtrail: chiavi ListPublic
	cloudtrail: ListQueries
	pista nuvolosa: ListTrails
	pista nuvolosa: LookupEvents
	cloudtrail: selettori PutEvent
	cloudtrail: selettori PutInsight
	cloudtrail: politica PutResource
	cloudtrail: RegisterOrganization DelegatedAdmin
	pista nuvolosa: RestoreEvent DataStore
	cloudtrail: Ingestione StartEvent DataStore
	pista nuvolosa: StartImport
	pista nuvolosa: StartLogging
	pista nuvolosa: StartQuery
	cloudtrail: Ingestione StopEvent DataStore
	pista nuvolosa: StopImport
	pista nuvolosa: StopLogging

Prefisso del servizio	Azioni
	pista nuvolosa: UpdateChannel pista nuvolosa: UpdateEvent DataStore pista nuvolosa: UpdateTrail

Prefisso del servizio	Azioni
cloudwatch	orologio cloud: DeleteAlarms cloudwatch: rilevatore DeleteAnomaly cloudwatch: DeleteDashboards cloudwatch: Regole DeleteInsight cloudwatch: Stream DeleteMetric cloudwatch: Storia DescribeAlarm cloudwatch: DescribeAlarms orologio nuvoloso: DescribeAlarms ForMetric cloudwatch: rilevatori DescribeAnomaly cloudwatch: Regole DescribeInsight cloudwatch: Azioni DisableAlarm cloudwatch: Regole DisableInsight cloudwatch: Azioni EnableAlarm cloudwatch: Regole EnableInsight cloudwatch: GetDashboard orologio nuvoloso: GetInsight RuleReport cloudwatch: Trasmissione in streaming GetMetric cloudwatch: ListDashboards orologio nuvoloso: ListManaged InsightRules cloudwatch: Stream ListMetric cloudwatch: rilevatore PutAnomaly

Prefisso del servizio	Azioni
	cloudwatch: Allarme PutComposite cloudwatch: PutDashboard cloudwatch: Regola PutInsight cloudwatch: PutManaged InsightRules cloudwatch: allarme PutMetric cloudwatch: Stream PutMetric cloudwatch: Stato SetAlarm cloudwatch: Stream StartMetric cloudwatch: Stream StopMetric

Prefisso del servizio	Azioni
codeartifact	codeartifact: Connessione AssociateExternal
	codeartifact: Versioni CopyPackage
	artefatto di codice: CreateDomain
	artefatto del codice: CreateRepository
	artefatto del codice: DeleteDomain
	artefatto del codice: DeleteDomain PermissionsPolicy
	artefatto del codice: DeletePackage
	codeartifact: Versioni DeletePackage
	artefatto di codice: DeleteRepository
	artefatto del codice: DeleteRepository PermissionsPolicy
	artefatto del codice: DescribeDomain
	artefatto del codice: DescribePackage
	codeartifact: versione DescribePackage
	artefatto di codice: DescribeRepository
	codeartifact: Connessione DisassociateExternal
	codeartifact: Versioni DisposePackage
	artefatto di codice: GetAssociated PackageGroup
	codeartifact: Token GetAuthorization
	artefatto di codice: GetDomain PermissionsPolicy
	artefatto del codice: GetPackage VersionAsset
	artefatto del codice: GetPackage VersionReadme

Prefisso del servizio	Azioni
	codeartifact: Punto finale GetRepository
	artefatto di codice: GetRepository PermissionsPolicy
	artefatto del codice: ListDomains
	codeartifact: Gruppi ListPackage
	artefatto di codice: ListPackages
	artefatto del codice: ListPackage VersionAssets
	artefatto del codice: ListPackage VersionDependencies
	codeartifact: versioni ListPackage
	artefatto di codice: ListRepositories
	artefatto del codice: ListRepositories InDomain
	codeartifact: versione PublishPackage
	artefatto di codice: PutDomain PermissionsPolicy
	codeartifact: Metadati PutPackage
	artefatto di codice: PutPackage OriginConfiguration
	artefatto del codice: PutRepository PermissionsPolicy
	codeartifact: Archivio ReadFrom
	artefatto di codice: UpdatePackage VersionsStatus
	artefatto del codice: UpdateRepository

Prefisso del servizio	Azioni
codedeploy	distribuzione del codice: BatchGet ApplicationRevisions codedeploy: Applicazioni BatchGet codedeploy: BatchGet DeploymentGroups codedeploy: BatchGet DeploymentInstances codedeploy: implementazioni BatchGet codedeploy: BatchGet DeploymentTargets codedeploy: Istanze BatchGet OnPremises codedeploy: ContinueDeployment codedeploy: CreateApplication codedeploy: CreateDeployment codedeploy: Config CreateDeployment codedeploy: Gruppo CreateDeployment codedeploy: DeleteApplication codedeploy: Config DeleteDeployment codedeploy: Gruppo DeleteDeployment codedeploy: Token DeleteGit HubAccount codedeploy: ID DeleteResources ByExternal codedeploy: DeregisterOn PremisesInstance codedeploy: GetApplication codedeploy: revisione GetApplication codedeploy: GetDeployment

Prefisso del servizio	Azioni
	codedeploy: Config GetDeployment codedeploy: Gruppo GetDeployment codedeploy: Istanza GetDeployment codedeploy: Target GetDeployment codedeploy: GetOn PremisesInstance codedeploy: revisioni ListApplication codedeploy: ListApplications codedeploy: configurazioni ListDeployment codedeploy: Gruppi ListDeployment codedeploy: Istanze ListDeployment codedeploy: ListDeployments codedeploy: obiettivi ListDeployment codedeploy: ListGit HubAccount TokenNames codedeploy: ListOn PremisesInstances codedeploy: PutLifecycle EventHook ExecutionStatus codedeploy: revisione RegisterApplication codedeploy: RegisterOn PremisesInstance codedeploy: SkipWait TimeFor InstanceTermination codedeploy: StopDeployment codedeploy: UpdateApplication codedeploy: Gruppo UpdateDeployment

Prefisso del servizio	Azioni
codeguru-profiler	codeguru-profiler: Canali AddNotification
	codeguru-profiler: Dati BatchGet FrameMetric
	codeguru-profiler: ConfigureAgent
	codeguru-profiler: Gruppo CreateProfiling
	codeguru-profiler: Gruppo DeleteProfiling
	codeguru-profiler: Gruppo DescribeProfiling
	codeguru-profiler: Riepilogo GetFindings ReportAccount
	codeguru-profiler: GetNotification Configurazione
	codeguru-profiler: GetPolicy
	codeguru profiler: GetProfile
	codeguru profiler: GetRecommendations
	codeguru-profiler: Rapporti ListFindings
	codeguru-profiler: Orari ListProfile
	codeguru-profiler: Gruppi ListProfiling
	codeguru-profiler: PutPermission
	codeguru-profiler: Canale RemoveNotification
	codeguru-profiler: RemovePermission
	codeguru profiler: SubmitFeedback
	codeguru-profiler: Gruppo UpdateProfiling

Prefisso del servizio	Azioni
codeguru-reviewer	revisore di codeguru: AssociateRepository codeguru-reviewer: recensione CreateCode codeguru-reviewer: Recensione DescribeCode codeguru-reviewer: Feedback DescribeRecommendation codeguru-reviewer: Associazione DescribeRepository codeguru-reviewer: DisassociateRepository codeguru-reviewer: Recensioni ListCode codeguru-reviewer: Feedback ListRecommendation revisore di codeguru: ListRecommendations codeguru-reviewer: Associazioni ListRepository codeguru-reviewer: Feedback PutRecommendation

Prefisso del servizio	Azioni
codepipeline	pipeline di codici: AcknowledgeJob
	pipeline di codice: AcknowledgeThird PartyJob
	pipeline di codice: CreateCustom ActionType
	pipeline di codice: CreatePipeline
	pipeline di codice: DeleteCustom ActionType
	pipeline di codice: DeletePipeline
	pipeline di codice: DeleteWebhook
	codepipeline: Festa DeregisterWebhook WithThird
	codepipeline: Tipo GetAction
	codepipeline: Dettagli GetJob
	codepipeline: GetPipeline
	codepipeline: esecuzione GetPipeline
	codepipeline: Stato GetPipeline
	codepipeline: Dettagli GetThird PartyJob
	codepipeline: Esecuzioni ListAction
	codepipeline: tipi ListAction
	codepipeline: Esecuzioni ListPipeline
	codepipeline: ListPipelines
	pipeline di codice: ListWebhooks
	codepipeline: Offerte di lavoro PollFor
	codepipeline: Offerte di lavoro PollFor ThirdParty

Prefisso del servizio	Azioni
	codepipeline: revisione PutAction
	codepipeline: Risultato PutApproval
	codepipeline: PutJob FailureResult
	pipeline di codice: PutJob SuccessResult
	pipeline di codice: PutThird PartyJob FailureResult
	pipeline di codice: PutThird PartyJob SuccessResult
	pipeline di codice: PutWebhook
	codepipeline: Festa RegisterWebhook WithThird
	codepipeline: RollbackStage
	codepipeline: esecuzione StartPipeline
	codepipeline: Esecuzione StopPipeline
	codepipeline: Tipo UpdateAction
	codepipeline: UpdatePipeline

Prefisso del servizio	Azioni
codestar	codestar: Membro AssociateTeam codestar: CreateProject codestar: Profilo CreateUser codestar: DeleteProject codestar: Profilo DeleteUser codestar: DescribeProject codestar: Profilo DescribeUser codestar: Membro DisassociateTeam codestar: ListProjects codestar: ListResources codestar: Membri ListTeam codestar: Profili ListUser codestar: UpdateProject codestar: Membro UpdateTeam codestar: Profilo UpdateUser

Prefisso del servizio	Azioni
codestar-notifications	codestar-notifications: Regola CreateNotification codestar-notification: regola DeleteNotification notifiche codestar: DeleteTarget codestar-notification: regola DescribeNotification codestar-notification: tipi ListEvent codestar-notification: regole ListNotification notifiche codestar: ListTargets codestar-notifications:Subscribe codestar-notifications:Unsubscribe codestar-notification: regola UpdateNotification

Prefisso del servizio	Azioni
cognito-identity	cognito-identity: Pool CreateIdentity identità cognitiva: DeleteIdentities identità cognitiva: Pool DeleteIdentity identità cognitiva: DescribeIdentity identità cognitiva: Pool DescribeIdentity identità cognitiva: GetIdentity PoolRoles identità cognitiva: ListIdentities cognito-identità: pool ListIdentity cognito-identità: identità LookupDeveloper cognito-identità: identità MergeDeveloper identità cognitiva: SetIdentity PoolRoles cognito-identità: identità UnlinkDeveloper cognito-identità: Pool UpdateIdentity

Prefisso del servizio	Azioni
cognito-idp	cognito-idp: attributi AddCustom
	cognito-idp: AdminAdd UserTo Gruppo
	cognito-idp: AdminConfirm SignUp
	cognito-idp: Utente AdminCreate
	cognito-idp: AdminDelete Utente
	cognito-idp: AdminDelete UserAttributes
	cognito-idp: Utente AdminDisable ProviderFor
	cognito-idp: AdminDisable Utente
	cognito-idp: AdminEnable Utente
	cognito-idp: AdminForget dispositivo
	cognito-idp: AdminGet Dispositivo
	cognito-idp: AdminGet Utente
	cognito-idp: AdminInitiate Autenticazione
	cognito-idp: AdminLink ProviderFor Utente
	cognito-idp: AdminList dispositivi
	cognito-idp: AdminList GroupsFor Utente
	cognito-idp: AdminList UserAuth Eventi
	cognito-idp: AdminRemove UserFrom Gruppo
	cognito-idp: AdminReset UserPassword
	cognito-idp: sfida AdminRespond ToAuth
	cognito-idp: AdminSet UserMFAPReference

Prefisso del servizio	Azioni
	cognito-idp: AdminSet UserPassword
	cognito-idp: AdminSet UserSettings
	cognito-idp: Feedback AdminUpdate AuthEvent
	cognito-idp: AdminUpdate DeviceStatus
	cognito-idp: AdminUpdate UserAttributes
	cognito-idp: Fuori AdminUser GlobalSign
	cognito-idp: AssociateSoftware Token
	cognito-idp: ChangePassword
	cognito-idp: ConfirmDevice
	cognito-idp: ConfirmForgot Parola d'ordine
	cognito-idp: ConfirmSign Attivo
	cognito-idp: CreateGroup
	cognito-idp: fornitore CreateIdentity
	cognito-idp: CreateResource server
	cognito-idp: CreateUser ImportJob
	cognito-idp: piscina CreateUser
	cognito-idp: CreateUser PoolClient
	cognito-idp: CreateUser PoolDomain
	cognito-idp: DeleteGroup
	cognito-idp: fornitore DeleteIdentity
	cognito-idp: DeleteResource server

Prefisso del servizio	Azioni
	cognito-idp: DeleteUser
	cognito-idp: attributi DeleteUser
	cognito-idp: DeleteUser piscina
	cognito-idp: DeleteUser PoolClient
	cognito-idp: DeleteUser PoolDomain
	cognito-idp: fornitore DescribeIdentity
	cognito-idp: DescribeResource server
	cognito-idp: DescribeRisk configurazione
	cognito-idp: DescribeUser ImportJob
	cognito-idp: piscina DescribeUser
	cognito-idp: DescribeUser PoolClient
	cognito-idp: DescribeUser PoolDomain
	cognito-idp: ForgetDevice
	cognito-idp: ForgotPassword
	cognito-idp:GetCSVHeader
	cognito-idp: GetDevice
	cognito-idp: GetGroup
	cognito-idp: GetIdentity ProviderBy identificatore
	cognito-idp: GetLog DeliveryConfiguration
	cognito-idp: certificato GetSigning
	cognito-idp:GetUICustomization

Prefisso del servizio	Azioni
	cognito-idp: GetUser
	cognito-idp: Codice GetUser AttributeVerification
	cognito-idp:Config GetUser PoolMfa
	cognito-idp: GlobalSign Fuori
	cognito-idp: InitiateAuth
	cognito-idp: ListDevices
	cognito-idp: ListGroups
	cognito-idp: fornitori ListIdentity
	cognito-idp: ListResource server
	cognito-idp: ListUser ImportJobs
	cognito-idp: ListUser PoolClients
	cognito-idp: piscine ListUser
	cognito-idp: ListUsers
	cognito-idp: ListUsers InGroup
	cognito-idp: Codice ResendConfirmation
	cognito-idp: RespondTo AuthChallenge
	cognito-idp: RevokeToken
	cognito-idp: SetLog DeliveryConfiguration
	cognito-idp: configurazione SetRisk
	cognito-idp:SetUICustomization
	cognito-idp: SetUser MFAPReference

Prefisso del servizio	Azioni
	cognito-idp:Config SetUser PoolMfa
	cognito-idp: SetUser Impostazioni
	cognito-idp: SignUp
	cognito-idp: StartUser ImportJob
	cognito-idp: StopUser ImportJob
	cognito-idp: UpdateAuth EventFeedback
	cognito-idp: UpdateDevice stato
	cognito-idp: UpdateGroup
	cognito-idp: fornitore UpdateIdentity
	cognito-idp: UpdateResource server
	cognito-idp: UpdateUser attributi
	cognito-idp: UpdateUser piscina
	cognito-idp: UpdateUser PoolClient
	cognito-idp: UpdateUser PoolDomain
	cognito-idp: VerifySoftware gettone
	cognito-idp: VerifyUser Attributo

Prefisso del servizio	Azioni
cognito-sync	sincronizzazione cognitiva: BulkPublish sincronizzazione cognitiva: DeleteDataset sincronizzazione cognitiva: DescribeDataset sincronizzazione cognitiva: DescribeIdentity PoolUsage cognito-sync: utilizzo DescribeIdentity cognito-sync: GetBulk PublishDetails cognito-sync: Eventi GetCognito cognito-sync: GetIdentity PoolConfiguration sincronizzazione cognitiva: ListDatasets sincronizzazione cognitiva: ListIdentity PoolUsage sincronizzazione cognitiva: ListRecords sincronizzazione cognitiva: RegisterDevice cognito-sync: Eventi SetCognito cognito-sync: SetIdentity PoolConfiguration cognito-sync: set di dati SubscribeTo cognito-sync: UnsubscribeFrom set di dati cognito-sync: UpdateRecords

Prefisso del servizio	Azioni
comprehendmedical	includi medicina: DetectionV2Job DescribeEntities ComprehendMedical: descrivi ICD 10 cm InferenceJob ComprehendMedical: descrivi il PHI DetectionJob comprehendmedical: Job DescribeRx NormInference ComprehendMedical: descrive NomeDCT InferenceJob comprende medicina: V2 DetectEntities comprehendmedical:DetectPHI comprehendmedical:InferICD10CM comprendi la medicina: Norma InferRx comprehendmedical:InferSNOMEDCT comprehendmedical: Detection V2 Jobs ListEntities Comprehend Medical: elenco D 10 cm InferenceJobs Comprehend Medical: Elenco PHI DetectionJobs comprehendmedical: Offerte di lavoro ListRx NormInference ComprehendMedical: elenca NomedCT InferenceJobs comprehendmedical: DetectionV2Job StartEntities ComprehendMedical: Start ICD 10 cm InferenceJob ComprehendMedical: Start PHI DetectionJob comprehendmedical: Job StartRx NormInference ComprehendMedical: avvia nomedCT InferenceJob comprehendmedical: DetectionV2Job StopEntities

Prefisso del servizio	Azioni
	ComprehendMedical: STOP ICD 10 cm InferenceJob Comprehend Medical: STOPPHI DetectionJob comprehendmedical: Job StopRx NormInference ComprehendMedical: ferma SnomedCT InferenceJob

Prefisso del servizio	Azioni
compute-optimizer	compute-optimizer: Preferenze DeleteRecommendation ottimizzatore di calcolo: DescribeRecommendation ExportJobs compute-optimizer: raccomandazioni ExportAuto ScalingGroup Ottimizzatore di calcolo: exportEBS VolumeRecommendations Ottimizzatore di calcolo: ExportEC2 InstanceRecommendations Ottimizzatore di calcolo: ExportECS ServiceRecommendations ottimizzatore di calcolo: ExportLambda FunctionRecommendations compute-optimizer: raccomandazioni ExportLicense Ottimizzatore di calcoloRecommendationProjected: getEC2 Metrics Ottimizzatore di calcolo: GETECS ServiceRecommendation ProjectedMetrics ottimizzatore di calcolo: GetEffective RecommendationPreferences ottimizzatore di calcolo: stato GetEnrollment compute-optimizer: Organizzazione GetEnrollment StatusesFor compute-optimizer: Preferenze GetRecommendation compute-optimizer: riassunti GetRecommendation compute-optimizer: PutRecommendation Preferenze compute-optimizer: Status UpdateEnrollment

Prefisso del servizio	Azioni
config	configurazione: BatchGet ResourceConfig
	config: Autorizzazione DeleteAggregation
	config: Regola DeleteConfig
	config: Aggregatore DeleteConfiguration
	config: Registratore DeleteConfiguration
	config: pacchetto DeleteConformance
	config: Canale DeleteDelivery
	config: Risultati DeleteEvaluation
	configurazione: DeleteOrganization ConfigRule
	configurazione: DeleteOrganization ConformancePack
	configurazione: DeletePending AggregationRequest
	config: configurazione DeleteRemediation
	config: eccezioni DeleteRemediation
	config: DeleteResource Config
	config: Configurazione DeleteRetention
	config: Interrogazione DeleteStored
	config: Istantanea DeliverConfig
	configurazione: DescribeAggregate ComplianceBy ConfigRules
	configurazione: DescribeAggregate ComplianceBy Conforman cePacks
	config: Autorizzazioni DescribeAggregation

Prefisso del servizio	Azioni
	config: Regola DescribeCompliance ByConfig
	configurazione: DescribeCompliance ByResource
	config: stato DescribeConfig RuleEvaluation
	config: Regole DescribeConfig
	config: Aggregatori DescribeConfiguration
	config: Stato DescribeConfiguration AggregatorSources
	config: Registratori DescribeConfiguration
	configurazione: DescribeConfiguration RecorderStatus
	configurazione: DescribeConformance PackCompliance
	config: pacchetti DescribeConformance
	configurazione: DescribeConformance PackStatus
	config: Canali DescribeDelivery
	configurazione: DescribeDelivery ChannelStatus
	configurazione: DescribeOrganization ConfigRules
	config: stati DescribeOrganization ConfigRule
	configurazione: DescribeOrganization ConformancePacks
	config: stati DescribeOrganization ConformancePack
	configurazione: DescribePending AggregationRequests
	config: configurazioni DescribeRemediation
	config: eccezioni DescribeRemediation
	configurazione: DescribeRemediation ExecutionStatus

Prefisso del servizio	Azioni
	config: configurazioni DescribeRetention
	configurazione: GetCompliance DetailsBy ConfigRule
	config: Risorsa GetCompliance DetailsBy
	configurazione: GetCompliance SummaryBy ConfigRule
	configurazione: GetCompliance SummaryBy ResourceType
	config: Dettagli GetConformance PackCompliance
	config: Riepilogo GetConformance PackCompliance
	configurazione: GetCustom RulePolicy
	configurazione: GetDiscovered ResourceCounts
	configurazione: GetOrganization ConfigRule DetailedStatus
	configurazione: GetOrganization ConformancePack DetailedStatus
	config: Politica GetOrganization CustomRule
	configurazione: GetResource ConfigHistory
	configurazione: GetResource EvaluationSummary
	config: Interrogazione GetStored
	config: Punteggi ListConformance PackCompliance
	config: Risorse ListDiscovered
	config: Valutazioni ListResource
	config: Interrogazioni ListStored
	config: Regola PutConfig
	config: Aggregatore PutConfiguration

Prefisso del servizio	Azioni
	config: Registratore PutConfiguration
	config: pacchetto PutConformance
	config: Canale PutDelivery
	configurazione: PutEvaluations
	config: Valutazione PutExternal
	configurazione: PutOrganization ConfigRule
	configurazione: PutOrganization ConformancePack
	config: configurazioni PutRemediation
	config: eccezioni PutRemediation
	config: PutResource Config
	config: Configurazione PutRetention
	config: Interrogazione PutStored
	config: SelectResource Config
	configurazione: StartConfig RulesEvaluation
	config: Registratore StartConfiguration
	config: Esecuzione StartRemediation
	config: Valutazione StartResource
	config: Recorder StopConfiguration

Prefisso del servizio	Azioni
connect	connect: Modulo ActivateEvaluation connetti: AssociateApproved Origin connettere: AssociateBot connect: AssociateDefault Vocabolario connettere: AssociateFlow connettere: AssociateInstance StorageConfig connect: AssociateLambda Funzione connect: AssociateLex Bot connetti: AssociatePhone NumberContact Flow connettere: AssociateQueue QuickConnects connettere: AssociateRouting ProfileQueues connetti: AssociateSecurity chiave connect: AssociateUser Competenze connettere: BatchGet FlowAssociation connetti: BatchPut Contatta connetti: ClaimPhone numero connect: CreateAgent stato connetti: CreateContact Flow connettere: CreateContact FlowModule connetti: CreateEvaluation Modulo connettere: CreateHours OfOperation

Prefisso del servizio	Azioni
	connettere: CreateInstance connect: CreateIntegration Associazione connettere: CreateParticipant connettere: CreatePersistent ContactAssociation connect: CreatePredefined Attributo connettere: CreatePrompt connettere: CreateQueue connetti: CreateQuick Connetti connect: CreateRouting Profilo connettere: CreateRule connect: CreateSecurity Profilo connect: CreateTask Modello connettere: CreateTraffic DistributionGroup connetti: CreateUse custodia connettere: CreateUser connettere: CreateUser HierarchyGroup connettere: CreateView connect: CreateView versione connettere: CreateVocabulary connetti: DeactivateEvaluation Modulo connect: DeleteContact valutazione

Prefisso del servizio	Azioni
	connect: DeleteContact Flow connettere: DeleteContact FlowModule connetti: DeleteEvaluation Modulo connettere: DeleteHours OfOperation connettere: DeleteInstance connect: DeleteIntegration Associazione connect: DeletePredefined Attributo connettere: DeletePrompt connettere: DeleteQueue connetti: DeleteQuick Connetti connect: DeleteRouting Profilo connettere: DeleteRule connect: DeleteSecurity Profilo connect: DeleteTask Modello connettere: DeleteTraffic DistributionGroup connetti: DeleteUse custodia connettere: DeleteUser connettere: DeleteUser HierarchyGroup connettere: DeleteView connettere: DeleteVocabulary connetti: DescribeAgent stato

Prefisso del servizio	Azioni
	connect: DescribeContact valutazione connect: DescribeContact Flow connettere: DescribeContact FlowModule connetti: DescribeEvaluation Modulo connect: DescribeInstance Attributo connettere: DescribeInstance StorageConfig connetti: DescribePhone numero connettere: DescribeRule connettere: DescribeTraffic DistributionGroup connettere: DescribeUser HierarchyGroup connettere: DescribeUser HierarchyStructure connettere: DescribeView connettere: DescribeVocabulary connettere: DisassociateApproved Origin connettere: DisassociateBot connettere: DisassociateFlow connettere: DisassociateInstance StorageConfig connect: DisassociateLambda Funzione connect: DisassociateLex Bot connetti: DisassociatePhone NumberContact Flow connettere: DisassociateQueue QuickConnects

Prefisso del servizio	Azioni
	connettere: DisassociateRouting ProfileQueues connetti: DisassociateSecurity chiave connect: DisassociateUser Competenze connect: Contatti DismissUser connect: GetContact Attributi connettere: GetCurrent MetricData connettere: GetCurrent UserData connetti: GetFederation Token connect: GetFlow Associazione connect: GetMetric Dati connessione: GetMetric dataV2 connect: File GetPrompt connect: GetTask modello connect: GetTraffic Distribuzione connect: ImportPhone numero connetti: ListApproved Origins connettere: ListBots connect: ListContact Valutazioni connettere: ListContact FlowModules connetti: ListContact Flows connect: ListContact Riferimenti

Prefisso del servizio	Azioni
	connect: ListDefault Vocabolari connect: Moduli ListEvaluation connettere: ListEvaluation FormVersions connect: ListFlow Associazioni connettere: ListHours OfOperations connect: ListInstance Attributi connettere: ListInstance StorageConfigs connect: ListIntegration Associazioni connect: ListLambda Funzioni connect: ListLex Bots connect: Numeri ListPhone connetti: ListPhone numbersV2 connect: Attributi ListPredefined connettere: ListPrompts connettere: ListQueue QuickConnects connettere: ListQueues connetti: ListQuick connette connetti: ListRealtime ContactAnalysis SegmentsV2 connettere: ListRouting ProfileQueues connect: ListRouting Profili connettere: ListRules

Prefisso del servizio	Azioni
	connect: ListSecurity chiavi connettere: ListSecurity ProfileApplications connettere: ListSecurity ProfilePermissions connect: ListSecurity Profili connect: ListTask Modelli connettere: ListTraffic DistributionGroups connect: ListUse Casi connettere: ListUser HierarchyGroups connect: ListUser Competenze connettere: ListUsers connettere: ListViews connect: ListView Versioni connettere: MonitorContact connettere: PauseContact connetti: PutUser stato connetti: ReleasePhone numero connettere: ReplicateInstance connettere: ResumeContact connetti: ResumeContact registrazione connettere: SearchAvailable PhoneNumbers connettere: SearchContacts

Prefisso del servizio	Azioni
	connettere: SearchHours OfOperations connect: SearchPredefined Attributi connettere: SearchPrompts connettere: SearchQueues connetti: SearchQuick connette connect: SearchRouting Profili connect: SearchSecurity Profili connettere: SearchVocabularies connettere: SendChat IntegrationEvent connetti: StartChat Contatta connect: StartContact valutazione connect: StartContact registrazione connessione: StartContact Streaming connettere: StartOutbound VoiceContact connetti: StartTask Contatta connetti: StartWeb RTCcontact connettere: StopContact connetti: StopContact registrazione connessione: StopContact Streaming connect: SubmitContact valutazione connect: SuspendContact registrazione

Prefisso del servizio	Azioni
	connettere: TransferContact connetti: UpdateAgent stato connettere: UpdateContact connect: UpdateContact Attributi connect: UpdateContact valutazione connettere: UpdateContact FlowContent connettere: UpdateContact FlowMetadata connetti: UpdateContact FlowModule contenuto connect: UpdateContact FlowModule Metadati connettere: UpdateContact FlowName connettere: UpdateContact RoutingData connetti: UpdateContact Pianifica connect: UpdateEvaluation Modulo connettere: UpdateHours OfOperation connect: UpdateInstance Attributo connettere: UpdateInstance StorageConfig connettere: UpdateParticipant RoleConfig connetti: UpdatePhone numero connettere: UpdatePhone NumberMetadata connect: UpdatePredefined Attributo connettere: UpdatePrompt

Prefisso del servizio	Azioni
	connetti: UpdateQueue HoursOf Funzionamento connettere: UpdateQueue MaxContacts connetti: UpdateQueue nome connect: UpdateQueue OutboundCaller Config connect: UpdateQueue stato connettere: UpdateQuick ConnectConfig connettere: UpdateQuick ConnectName connettere: UpdateRouting ProfileAgent AvailabilityTimer connettere: UpdateRouting ProfileConcurrency connettere: UpdateRouting ProfileDefault OutboundQueue connettere: UpdateRouting ProfileName connettere: UpdateRouting ProfileQueues connettere: UpdateRule connect: UpdateSecurity Profilo connect: UpdateTask Modello connect: UpdateTraffic Distribuzione connect: UpdateUser gerarchia connect: Nome UpdateUser HierarchyGroup connettere: UpdateUser HierarchyStructure connettere: UpdateUser IdentityInfo connettere: UpdateUser PhoneConfig

Prefisso del servizio	Azioni
	connect: UpdateUser Competenze connettere: UpdateUser RoutingProfile connettere: UpdateUser SecurityProfiles connetti: UpdateView contenuto connect: UpdateView Metadati
cur	cur: Definizione DeleteReport cur: Definizioni DescribeReport cur: Definizione ModifyReport cur: Definizione PutReport

Prefisso del servizio	Azioni
databrew	data brew: BatchDelete RecipeVersion data brew: CreateDataset databrew: Job CreateProfile datasrew: CreateProject data brew: CreateRecipe databrew: Job CreateRecipe datasrew: CreateRuleset data brew: CreateSchedule data brew: DeleteDataset data brew: DeleteJob data brew: DeleteProject databrew: versione DeleteRecipe databrew: DeleteRuleset data brew: DeleteSchedule data brew: DescribeDataset data brew: DescribeJob databrew: Esegui DescribeJob databrew: DescribeProject data brew: DescribeRecipe data brew: DescribeRuleset data brew: DescribeSchedule

Prefisso del servizio	Azioni
	data brew: ListDatasets databrew: esegue ListJob databrew: ListJobs data brew: ListProjects data brew: ListRecipes databrew: Versioni ListRecipe databrew: ListRulesets data brew: ListSchedules data brew: PublishRecipe data brew: SendProject SessionAction databrew: Esegui StartJob databrew: Sessione StartProject databrew: Esegui StopJob databrew: UpdateDataset databrew: Job UpdateProfile datasrew: UpdateProject data brew: UpdateRecipe databrew: Job UpdateRecipe datasrew: UpdateRuleset data brew: UpdateSchedule

Prefisso del servizio	Azioni
dataexchange	scambio di dati: CancelJob
	scambio di dati: impostato CreateData
	scambio di dati: Azione CreateEvent
	scambio di dati: CreateJob
	scambio di dati: CreateRevision
	scambio di dati: DeleteAsset
	scambio di dati: azione DeleteEvent
	scambio di dati: DeleteRevision
	scambio di dati: azione GetEvent
	scambio di dati: GetJob
	scambio di dati: ListData SetRevisions
	scambio di dati: insiemi ListData
	scambio di dati: Azioni ListEvent
	scambio di dati: ListJobs
	scambio di dati: risorse ListRevision
	scambio di dati: RevokeRevision
	scambio di dati: SendData SetNotification
	scambio di dati: StartJob
	scambio di dati: UpdateAsset
	scambio di dati: impostato UpdateData
	scambio di dati: Azione UpdateEvent

Prefisso del servizio	Azioni
	scambio di dati: UpdateRevision
datapipeline	pipeline dati: ActivatePipeline tubazione dati: CreatePipeline tubazione dati: DeactivatePipeline tubazione dati: DeletePipeline tubazione dati: DescribeObjects tubazione dati: DescribePipelines tubazione dati: EvaluateExpression datapipeline: definizione GetPipeline pipeline di dati: ListPipelines datapipeline: Attività PollFor datapipeline: definizione PutPipeline pipeline di dati: QueryObjects datapipeline: Progresso ReportTask pipeline dati: ReportTask RunnerHeartbeat tubazione dati: SetStatus datapipeline: Stato SetTask datapipeline: definizione ValidatePipeline

Prefisso del servizio	Azioni
dax	dax: CreateCluster dax: Fattore DecreaseReplication dax: DeleteCluster dax: Gruppo DeleteParameter dax: Gruppo DeleteSubnet dax: DescribeClusters dax: parametri DescribeDefault dax: DescribeEvents dax: Gruppi DescribeParameter dax: DescribeParameters dax: Gruppi DescribeSubnet dax: Fattore IncreaseReplication dax: RebootNode fax: UpdateCluster dax: Gruppo UpdateParameter dax: Gruppo UpdateSubnet

Prefisso del servizio	Azioni
devicefarm	devicefarm: Pool CreateDevice devicefarm: Profilo CreateInstance devicefarm: Profilo CreateNetwork devicefarm: CreateProject fabbrica di dispositivi: CreateRemote AccessSession fabbrica di dispositivi: CreateTest GridProject fabbrica di dispositivi: CreateTest GridUrl fabbrica di dispositivi: CreateUpload devicefarm:CreateVPCEConfiguration devicefarm: Piscina DeleteDevice devicefarm: Profilo DeleteInstance devicefarm: Profilo DeleteNetwork devicefarm: DeleteProject fabbrica di dispositivi: DeleteRemote AccessSession fabbrica di dispositivi: DeleteRun fabbrica di dispositivi: DeleteTest GridProject fabbrica di dispositivi: DeleteUpload devicefarm:DeleteVPCEConfiguration devicefarm: Impostazioni GetAccount devicefarm: GetDevice devicefarm: Istanza GetDevice

Prefisso del servizio	Azioni
	devicefarm: Pool GetDevice devicefarm: GetDevice PoolCompatibility devicefarm: Profilo GetInstance devicefarm: GetJob devicefarm: Profilo GetNetwork devicefarm: Stato GetOffering devicefarm: GetProject fabbrica di dispositivi: GetRemote AccessSession fabbrica di dispositivi: GetRun fattoria di dispositivi: GetSuite fattoria di dispositivi: GetTest fabbrica di dispositivi: GetTest GridProject fabbrica di dispositivi: GetTest GridSession fabbrica di dispositivi: GetUpload devicefarm:GetVPCEConfiguration fattoria di dispositivi: ListArtifacts devicefarm: Istanze ListDevice devicefarm: ListDevice Pool devicefarm: ListDevices devicefarm: Profili ListInstance devicefarm: ListJobs

Prefisso del servizio	Azioni
	devicefarm: Profili ListNetwork devicefarm: Promozioni ListOffering devicefarm: ListOfferings devicefarm: Transazioni ListOffering devicefarm: ListProjects fabbrica di dispositivi: ListRemote AccessSessions fabbrica di dispositivi: ListRuns fattoria di dispositivi: ListSamples fattoria di dispositivi: ListSuites fabbrica di dispositivi: ListTest GridProjects devicefarm: Azioni ListTest GridSession devicefarm: artefatti ListTest GridSession devicefarm: ListTest GridSessions fabbrica di dispositivi: ListTests devicefarm: problemi ListUnique devicefarm: ListUploads devicefarm:ListVPCEConfigurations fattoria di dispositivi: PurchaseOffering fattoria di dispositivi: RenewOffering fattoria di dispositivi: ScheduleRun fattoria di dispositivi: StopJob

Prefisso del servizio	Azioni
	fabbrica di dispositivi: StopRemote AccessSession fabbrica di dispositivi: StopRun devicefarm: Istanza UpdateDevice devicefarm: Pool UpdateDevice devicefarm: Profilo UpdateInstance devicefarm: Profilo UpdateNetwork devicefarm: UpdateProject fabbrica di dispositivi: UpdateTest GridProject fabbrica di dispositivi: UpdateUpload devicefarm:UpdateVPCEConfiguration

Prefisso del servizio	Azioni
devops-guru	devops-guru: Canale AddNotification
	devops-guru: DeleteInsight
	devops-guru: Health DescribeAccount
	devops-guru: DescribeAccount Panoramica
	devops-guru: DescribeAnomaly
	devops-guru: DescribeEvent SourcesConfig
	devops-guru: DescribeFeedback
	devops-guru: DescribeInsight
	devops-guru: Health DescribeOrganization
	devops-guru: DescribeOrganization Panoramica
	devops-guru: Health DescribeOrganization ResourceCollection
	devops-guru: DescribeResource CollectionHealth
	devops-guru: integrazione DescribeService
	devops-guru: GetCost stima
	devops-guru: GetResource Collezione
	devops-guru: ListAnomalies ForInsight
	devops-guru: ListAnomalous LogGroups
	devops-guru: ListEvents
	devops-guru: ListInsights
	devops-guru: Risorse ListMonitored
	devops-guru: ListNotification Canali

Prefisso del servizio	Azioni
	devops-guru: ListOrganization Approfondimenti
	devops-guru: ListRecommendations
	devops-guru: PutFeedback
	devops-guru: RemoveNotification Canale
	devops-guru: SearchInsights
	devops-guru: Approfondimenti SearchOrganization
	devops-guru: StartCost stima
	devops-guru: UpdateEvent SourcesConfig
	devops-guru: Collezione UpdateResource
	devops-guru: UpdateService Integrazione

Prefisso del servizio	Azioni
directconnect	connessione diretta: AcceptDirect ConnectGateway AssociationProposal connessione diretta: AllocateConnection OnInterconnect directconnect: connessione AllocateHosted connessione diretta: AllocatePrivate VirtualInterface connessione diretta: AllocatePublic VirtualInterface connessione diretta: AllocateTransit VirtualInterface connessione diretta: AssociateConnection WithLag directconnect: connessione AssociateHosted connessione diretta: AssociateMac SecKey directconnect: interfaccia AssociateVirtual connessione diretta: ConfirmConnection directconnect: accordo ConfirmCustomer connessione diretta: ConfirmPrivate VirtualInterface connessione diretta: ConfirmPublic VirtualInterface connessione diretta: ConfirmTransit VirtualInterface directconnect:CreateBGPPeer connessione diretta: CreateConnection connessione diretta: CreateDirect ConnectGateway directconnect: Associazione CreateDirect ConnectGateway connessione diretta: CreateDirect ConnectGateway AssociationProposal

Prefisso del servizio	Azioni
	connessione diretta: CreateInterconnect connessione diretta: CreateLag connessione diretta: CreatePrivate VirtualInterface connessione diretta: CreatePublic VirtualInterface connessione diretta: CreateTransit VirtualInterface directconnect:DeleteBGPPeer connessione diretta: DeleteConnection connessione diretta: DeleteDirect ConnectGateway directconnect: Associazione DeleteDirect ConnectGateway connessione diretta: DeleteDirect ConnectGateway Associati onProposal connessione diretta: DeleteInterconnect connessione diretta: DeleteLag directconnect: interfaccia DeleteVirtual directconnect: Loa DescribeConnection connessione diretta: DescribeConnections connessione diretta: DescribeConnections OnInterconnect directconnect: metadati DescribeCustomer connessione diretta: DescribeDirect ConnectGateway Associati onProposals directconnect: associazioni DescribeDirect ConnectGateway directconnect: Allegati DescribeDirect ConnectGateway

Prefisso del servizio	Azioni
	connessione diretta: DescribeDirect ConnectGateways directconnect: Connessioni DescribeHosted directconnect: Loa DescribeInterconnect connessione diretta: DescribeInterconnects connessione diretta: DescribeLags connessione diretta: DescribeLoa connessione diretta: DescribeLocations directconnect: configurazione DescribeRouter directconnect: gateway DescribeVirtual directconnect: Interfacce DescribeVirtual connessione diretta: DisassociateConnection FromLag connessione diretta: DisassociateMac SecKey directconnect: Storia ListVirtual InterfaceTest connessione diretta: StartBgp FailoverTest connessione diretta: StopBgp FailoverTest connessione diretta: UpdateConnection connessione diretta: UpdateDirect ConnectGateway directconnect: Associazione UpdateDirect ConnectGateway connessione diretta: UpdateLag connessione diretta: UpdateVirtual InterfaceAttributes

Prefisso del servizio	Azioni
dlm	dlm: Politica CreateLifecycle
	dlm: Politica DeleteLifecycle
	dlm: Politiche GetLifecycle
	dlm: Politica GetLifecycle
	dlm: Politica UpdateLifecycle

Prefisso del servizio	Azioni
dms	dms: ApplyPending MaintenanceAction dms: raccomandazioni BatchStart dms: Esegui CancelReplication TaskAssessment dms: fornitore CreateData dms: CreateEndpoint dms: abbonamento CreateEvent dms: Profilo CreateInstance dms: Progetto CreateMigration dms: Config CreateReplication dms: Istanza CreateReplication dms: CreateReplication SubnetGroup dms: Attività CreateReplication dms: DeleteCertificate dms: DeleteConnection dms: fornitore DeleteData dms: DeleteEndpoint dms: abbonamento DeleteEvent dms: DeleteFleet AdvisorCollector dms: DeleteFleet AdvisorDatabases dms: Profilo DeleteInstance dms: Progetto DeleteMigration

Prefisso del servizio	Azioni
	dms: Config DeleteReplication dms: Istanza DeleteReplication dms: DeleteReplication SubnetGroup dms: Attività DeleteReplication dms: Esegui DeleteReplication TaskAssessment dms: attributi DescribeAccount dms: DescribeApplicable IndividualAssessments dms: DescribeCertificates dms: DescribeConnections dms: DescribeEndpoints dms: Impostazioni DescribeEndpoint dms: tipi DescribeEndpoint dms: Versioni DescribeEngine dms: Categorie DescribeEvent dms: DescribeEvents dms: abbonamenti DescribeEvent dms: DescribeFleet AdvisorCollectors dms: DescribeFleet AdvisorDatabases dms: Analisi DescribeFleet AdvisorLsa dms: DescribeFleet AdvisorSchema ObjectSummary dms: DescribeFleet AdvisorSchemas

Prefisso del servizio	Azioni
	dms: DescribeMetadata ModelImports dms: DescribeOrderable ReplicationInstances dms: DescribePending MaintenanceActions dms: limitazioni DescribeRecommendation dms: DescribeRecommendations dms: DescribeRefresh SchemasStatus dms: configurazioni DescribeReplication dms: Istanze DescribeReplication dms: registri DescribeReplication InstanceTask dms: DescribeReplications dms: DescribeReplication SubnetGroups dms: DescribeReplication TableStatistics dms: risultati DescribeReplication TaskAssessment dms: Esegue DescribeReplication TaskAssessment dms: Valutazioni DescribeReplication TaskIndividual dms: Attività DescribeReplication dms: DescribeSchemas dms: Statistiche DescribeTable dms: ExportMetadata ModelAssessment dms: Modello GetMetadata dms: ImportCertificate

Prefisso del servizio	Azioni
	dms: ListMetadata ModelAssessment ActionItems dms: ModifyEndpoint dms: abbonamento ModifyEvent dms: Config ModifyReplication dms: Istanza ModifyReplication dms: ModifyReplication SubnetGroup dms: Attività ModifyReplication dms: Attività MoveReplication dms: Istanza RebootReplication dms: RefreshSchemas dms: tabelle ReloadReplication dms: ReloadTables dms: Analisi RunFleet AdvisorLsa dms: StartMetadata ModelAssessment dms: StartMetadata ModelConversion dms: StartMetadata ModelExport ToTarget dms: StartRecommendations dms: StartReplication dms: Attività StartReplication dms: StartReplication TaskAssessment dms: Attività StopReplication

Prefisso del servizio	Azioni
	dms: TestConnection dms: ponte UpdateSubscriptions ToEvent
docdb-elastic	docdb-elastic: istantanea CopyCluster docdb-elastico: DeleteCluster docdb-elastic: istantanea DeleteCluster docdb-elastico: GetCluster docdb-elastic: istantanea GetCluster docdb-elastico: ListClusters docdb-elastic: istantanee ListCluster docdb-elastico: RestoreCluster FromSnapshot docdb elastico: StartCluster docdb elastico: StopCluster docdb elastico: UpdateCluster

Prefisso del servizio	Azioni
ds	ds: cartella AcceptShared ds: AddIp percorsi Annunci: AddRegion ds: CancelSchema estensione Annunci: ConnectDirectory annunci: CreateAlias annunci: CreateComputer ds: CreateConditional spedizioniere Annunci: CreateDirectory ds: CreateLog abbonamento Annunci: CreateMicrosoft AD Annunci: CreateSnapshot annunci: CreateTrust ds: DeleteConditional spedizioniere Annunci: DeleteDirectory ds: DeleteLog abbonamento Annunci: DeleteSnapshot annunci: DeleteTrust annunci: DeregisterCertificate ds: DeregisterEvent Argomento annunci: DescribeCertificate

Prefisso del servizio	Azioni
	annunci: DescribeClient AuthenticationSettings ds: DescribeConditional spedizionieri Annunci: DescribeDirectories ds: DescribeDomain Controller ds: DescribeEvent Argomenti ds:DescribeLDAPSSettings annunci: DescribeRegions annunci: DescribeSettings ds: DescribeShared elenchi annunci: DescribeSnapshots annunci: DescribeTrusts ds: DescribeUpdate Elenco ds: DisableClient autenticazione ds:DisableLDAPS annunci: DisableRadius annunci: DisableSso ds: EnableClient autenticazione ds:EnableLDAPS annunci: EnableRadius annunci: EnableSso ds: GetDirectory limiti

Prefisso del servizio	Azioni
	ds: GetSnapshot Limiti Annunci: ListCertificates ds: ListIp Percorsi ds: ListLog abbonamenti ds: estensioni ListSchema annunci: RegisterCertificate ds: RegisterEvent Argomento ds: RejectShared Elenco ds: RemoveIp percorsi Annunci: RemoveRegion ds: ResetUser password ds: RestoreFrom istantanea annunci: ShareDirectory ds: StartSchema estensione Annunci: UnshareDirectory ds: UpdateConditional spedizioniere ds: Configurazione UpdateDirectory ds: UpdateNumber OfDomain Controller Annunci: UpdateRadius annunci: UpdateSettings annunci: UpdateTrust

Prefisso del servizio	Azioni
	annunci: VerifyTrust

Prefisso del servizio	Azioni
dynamodb	dinamodb: CreateBackup
	dynamodb: Tabella CreateGlobal
	dynamodb: CreateTable
	dinamodb: DeleteBackup
	dinamodb: DeleteTable
	dinamodb: DescribeBackup
	dynamodb: backup DescribeContinuous
	dynamodb: Approfondimenti DescribeContributor
	dynamodb: DescribeEndpoints
	dinamodb: DescribeExport
	dynamodb: Tabella DescribeGlobal
	dynamodb: DescribeGlobal TableSettings
	dinamodb: DescribeImport
	dinamodb: DescribeKinesis StreamingDestination
	dinamodb: DescribeLimits
	dinamodb: DescribeStream
	dinamodb: DescribeTable
	dynamodb: Scalabilità DescribeTable ReplicaAuto
	dynamodb: DescribeTime ToLive
	dinamodb: DisableKinesis StreamingDestination
	dinamodb: EnableKinesis StreamingDestination

Prefisso del servizio	Azioni
	dinamodb: ExportTable ToPoint InTime
	dynamodb: politica GetResource
	dynamodb: ImportTable
	dinamodb: ListBackups
	dynamodb: Approfondimenti ListContributor
	dynamodb: ListExports
	dynamodb: tabelle ListGlobal
	dynamodb: ListImports
	dinamodb: ListStreams
	dinamodb: ListTables
	dinamodb: RestoreTable FromBackup
	dinamodb: RestoreTable ToPoint InTime
	dynamodb: backup UpdateContinuous
	dynamodb: Approfondimenti UpdateContributor
	dynamodb: Tabella UpdateGlobal
	dynamodb: UpdateGlobal TableSettings
	dinamodb: UpdateKinesis StreamingDestination
	dinamodb: UpdateTable
	dynamodb: Scalabilità UpdateTable ReplicaAuto
	dynamodb: UpdateTime ToLive

Prefisso del servizio	Azioni
ebs	Web: CompleteSnapshot ebs: StartSnapshot

Prefisso del servizio	Azioni
ec2	ec2: Trasferimento AcceptAddress
	ec2: Citazione AcceptReserved InstancesExchange
	ec2: AcceptTransit GatewayMulticast DomainAssociations
	ec2: Allegato AcceptTransit GatewayPeering
	ec2: Allegato AcceptTransit GatewayVpc
	ec2: AcceptVpc EndpointConnections
	ec2: AcceptVpc PeeringConnection
	ec2: Sidro AdvertiseByoip
	ec2: AllocateAddress
	ec2: AllocateHosts
	ec2: AllocateIpam PoolCidr
	ec2: ApplySecurity GroupsTo ClientVpn TargetNetwork
	ec2:6 indirizzi AssignIpv
	ec2: AssignPrivate IpAddresses
	ec2: Indirizzo AssignPrivate NatGateway
	ec2: AssociateAddress
	ec2: Rete AssociateClient VpnTarget
	ec2: Opzioni AssociateDhcp
	ec2: Ruolo AssociateEnclave CertificateIam
	ec2: AssociateIam InstanceProfile
	ec2: AssociateInstance EventWindow

Prefisso del servizio	Azioni
	ec2: Byosan AssociateIpam
	ec2: AssociateIpam ResourceDiscovery
	ec2: AssociateNat GatewayAddress
	ec2: Tabella AssociateRoute
	ec2: AssociateSubnet CidrBlock
	ec2: Dominio AssociateTransit GatewayMulticast
	ec2: Tabella AssociateTransit GatewayPolicy
	ec2: Tabella AssociateTransit GatewayRoute
	ec2: Interfaccia AssociateTrunk
	ec2: AssociateVpc CidrBlock
	ec2: AttachClassic LinkVpc
	ec2: porta AttachInternet
	ec2: Interfaccia AttachNetwork
	ec2: Fornitore AttachVerified AccessTrust
	ec2: AttachVolume
	ec2: porta AttachVpn
	ec2: AuthorizeClient VpnIngress
	ec2: AuthorizeSecurity GroupEgress
	ec2: AuthorizeSecurity GroupIngress
	ec2: BundleInstance
	ec2: Attività CancelBundle

Prefisso del servizio	Azioni
	ec2: Prenotazione CancelCapacity
	ec2: CancelCapacity ReservationFleets
	ec2: Attività CancelConversion
	ec2: Attività CancelExport
	ec2: CancelImage LaunchPermission
	ec2: Attività CancelImport
	ec2: CancelReserved InstancesListing
	ec2: CancelSpot FleetRequests
	ec2: CancelSpot InstanceRequests
	ec2: Istanza ConfirmProduct
	ec2: Immagine CopyFpga
	ec2: CopyImage
	ec2: CopySnapshot
	ec2: Prenotazione CreateCapacity
	ec2: CreateCapacity ReservationFleet
	ec2: porta CreateCarrier
	ec2: CreateClient VpnEndpoint
	ec2: CreateClient VpnRoute
	ec2: Sidro CreateCoip
	ec2: Piscina CreateCoip
	ec2: Gateway CreateCustomer

Prefisso del servizio	Azioni
	ec2: sottorete CreateDefault
	ec2: Vpc CreateDefault
	ec2: Opzioni CreateDhcp
	ec2: Gateway CreateEgress OnlyInternet
	ec2: CreateFleet
	ec2: registri CreateFlow
	ec2: Immagine CreateFpga
	ec2: CreateImage
	ec2: CreateInstance ConnectEndpoint
	ec2: CreateInstance EventWindow
	ec2: CreateInstance ExportTask
	ec2: porta CreateInternet
	ec2: CreateIpam
	ec2: Piscina CreateIpam
	ec2: CreateIpam ResourceDiscovery
	ec2: Ambito CreateIpam
	ec2: coppia CreateKey
	ec2: modello CreateLaunch
	ec2: CreateLaunch TemplateVersion
	ec2: CreateLocal GatewayRoute
	ec2: Tabella CreateLocal GatewayRoute

Prefisso del servizio	Azioni
	ec2: Associazione CreateLocal GatewayRoute TableVirtual InterfaceGroup
	ec2: Associazione CreateLocal GatewayRoute TableVpc
	ec2: CreateManaged PrefixList
	ec2: porta CreateNat
	ec2: Acl CreateNetwork
	ec2: CreateNetwork AclEntry
	ec2: Ambito CreateNetwork InsightsAccess
	ec2: CreateNetwork InsightsPath
	ec2: Interfaccia CreateNetwork
	ec2: CreateNetwork InterfacePermission
	ec2: Gruppo CreatePlacement
	ec2: Pool IPv4 CreatePublic
	CreateReplaceRootVolumeec2: Attività
	ec2: CreateReserved InstancesListing
	ec2: CreateRestore ImageTask
	ec2: CreateRoute
	ec2: Tabella CreateRoute
	ec2: Gruppo CreateSecurity
	ec2: CreateSnapshot
	ec2: CreateSnapshots

Prefisso del servizio	Azioni
	ec2: CreateSpot DatafeedSubscription
	ec2: CreateStore ImageTask
	ec2: CreateSubnet
	ec2: CreateSubnet CidrReservation
	ec2: CreateTraffic MirrorFilter
	ec2: Regola CreateTraffic MirrorFilter
	ec2: CreateTraffic MirrorSession
	ec2: CreateTraffic MirrorTarget
	ec2: porta CreateTransit
	ec2: CreateTransit GatewayConnect
	ec2: Peer CreateTransit GatewayConnect
	ec2: Dominio CreateTransit GatewayMulticast
	ec2: Allegato CreateTransit GatewayPeering
	ec2: Tabella CreateTransit GatewayPolicy
	ec2: CreateTransit GatewayPrefix ListReference
	ec2: CreateTransit GatewayRoute
	ec2: Tabella CreateTransit GatewayRoute
	ec2: CreateTransit GatewayRoute TableAnnouncement
	ec2: Allegato CreateTransit GatewayVpc
	ec2: CreateVerified AccessEndpoint
	ec2: CreateVerified AccessGroup

Prefisso del servizio	Azioni
	ec2: CreateVerified AccessInstance
	ec2: Fornitore CreateVerified AccessTrust
	ec2: CreateVolume
	ec2: CreateVpc
	ec2: Punto finale CreateVpc
	ec2: Notifica CreateVpc EndpointConnection
	ec2: Configurazione CreateVpc EndpointService
	ec2: CreateVpc PeeringConnection
	ec2: Connessione CreateVpn
	ec2: CreateVpn ConnectionRoute
	ec2: porta CreateVpn
	ec2: Porta DeleteCarrier
	ec2: DeleteClient VpnEndpoint
	ec2: DeleteClient VpnRoute
	ec2: Sidro DeleteCoip
	ec2: Piscina DeleteCoip
	ec2: Gateway DeleteCustomer
	ec2: Opzioni DeleteDhcp
	ec2: Gateway DeleteEgress OnlyInternet
	ec2: DeleteFleets
	ec2: registri DeleteFlow

Prefisso del servizio	Azioni
	ec2: Immagine DeleteFpga
	ec2: DeleteInstance ConnectEndpoint
	ec2: DeleteInstance EventWindow
	ec2: porta DeleteInternet
	ec2: DeleteIpam
	ec2: Piscina DeleteIpam
	ec2: DeleteIpam ResourceDiscovery
	ec2: Ambito DeleteIpam
	ec2: coppia DeleteKey
	ec2: modello DeleteLaunch
	ec2: DeleteLaunch TemplateVersions
	ec2: DeleteLocal GatewayRoute
	ec2: Tabella DeleteLocal GatewayRoute
	ec2: Associazione DeleteLocal GatewayRoute TableVirtual InterfaceGroup
	ec2: Associazione DeleteLocal GatewayRoute TableVpc
	ec2: DeleteManaged PrefixList
	ec2: porta DeleteNat
	ec2: Acl DeleteNetwork
	ec2: DeleteNetwork AclEntry
	ec2: Ambito DeleteNetwork InsightsAccess

Prefisso del servizio	Azioni
	ec2: DeleteNetwork InsightsAccess ScopeAnalysis
	ec2: DeleteNetwork InsightsAnalysis
	ec2: DeleteNetwork InsightsPath
	ec2: Interfaccia DeleteNetwork
	ec2: DeleteNetwork InterfacePermission
	ec2: Gruppo DeletePlacement
	ec2: Pool IPv4 DeletePublic
	ec2: DeleteQueued ReservedInstances
	ec2: DeleteRoute
	ec2: Tabella DeleteRoute
	ec2: Gruppo DeleteSecurity
	ec2: DeleteSnapshot
	ec2: DeleteSpot DatafeedSubscription
	ec2: DeleteSubnet
	ec2: DeleteSubnet CidrReservation
	ec2: DeleteTraffic MirrorFilter
	ec2: Regola DeleteTraffic MirrorFilter
	ec2: DeleteTraffic MirrorSession
	ec2: DeleteTraffic MirrorTarget
	ec2: porta DeleteTransit
	ec2: DeleteTransit GatewayConnect

Prefisso del servizio	Azioni
	ec2: Peer DeleteTransit GatewayConnect
	ec2: Dominio DeleteTransit GatewayMulticast
	ec2: Allegato DeleteTransit GatewayPeering
	ec2: Tabella DeleteTransit GatewayPolicy
	ec2: DeleteTransit GatewayPrefix ListReference
	ec2: DeleteTransit GatewayRoute
	ec2: Tabella DeleteTransit GatewayRoute
	ec2: DeleteTransit GatewayRoute TableAnnouncement
	ec2: Allegato DeleteTransit GatewayVpc
	ec2: DeleteVerified AccessEndpoint
	ec2: DeleteVerified AccessGroup
	ec2: DeleteVerified AccessInstance
	ec2: Fornitore DeleteVerified AccessTrust
	ec2: DeleteVolume
	ec2: DeleteVpc
	ec2: Notifiche DeleteVpc EndpointConnection
	ec2: Endpoint DeleteVpc
	ec2: Configurazioni DeleteVpc EndpointService
	ec2: DeleteVpc PeeringConnection
	ec2: Connessione DeleteVpn
	ec2: DeleteVpn ConnectionRoute

Prefisso del servizio	Azioni
	ec2: porta DeleteVpn
	ec2: Sidro DeprovisionByoip
	ec2: Byosan DeprovisionIpam
	ec2: DeprovisionIpam PoolCidr
	ec2: IPv4 DeprovisionPublic PoolCidr
	ec2: DeregisterImage
	ec2: attributi DeregisterInstance EventNotification
	ec2: DeregisterTransit GatewayMulticast GroupMembers
	ec2: DeregisterTransit GatewayMulticast GroupSources
	ec2: Attributi DescribeAccount
	ec2: DescribeAddresses
	ec2: Attributo DescribeAddresses
	ec2: Trasferimenti DescribeAddress
	ec2: DescribeAggregate IdFormat
	ec2: Zone DescribeAvailability
	ec2: DescribeAws NetworkPerformance MetricSubscriptions
	ec2: Attività DescribeBundle
	ec2: Sidri DescribeByoip
	ec2: DescribeCapacity ReservationFleets
	ec2: Prenotazioni DescribeCapacity
	ec2: Gateway DescribeCarrier

Prefisso del servizio	Azioni
	ec2: DescribeClassic LinkInstances
	ec2: Regole DescribeClient VpnAuthorization
	ec2: DescribeClient VpnConnections
	ec2: DescribeClient VpnEndpoints
	ec2: DescribeClient VpnRoutes
	ec2: Reti DescribeClient VpnTarget
	ec2: Piscine DescribeCoip
	ec2: Attività DescribeConversion
	ec2: Gateway DescribeCustomer
	ec2: Opzioni DescribeDhcp
	ec2: Gateway DescribeEgress OnlyInternet
	ec2: GPU DescribeElastic
	ec2: DescribeExport ImageTasks
	ec2: Attività DescribeExport
	ec2: DescribeFast LaunchImages
	ec2: DescribeFast SnapshotRestores
	ec2: Storia DescribeFleet
	ec2: Istanze DescribeFleet
	ec2: DescribeFleets
	ec2: registri DescribeFlow
	ec2: DescribeFpga ImageAttribute

Prefisso del servizio	Azioni
	ec2: immagini DescribeFpga
	ec2: DescribeHost ReservationOfferings
	ec2: Prenotazioni DescribeHost
	ec2: DescribeHosts
	ec2: Associazioni DescribeIam InstanceProfile
	ec2: DescribeIdentity IdFormat
	ec2: Formato DescribeId
	ec2: Attributo DescribeImage
	ec2: DescribeImages
	ec2: DescribeImport ImageTasks
	ec2: DescribeImport SnapshotTasks
	ec2: Attributo DescribeInstance
	ec2: DescribeInstance ConnectEndpoints
	ec2: DescribeInstance CreditSpecifications
	ec2: Attributi DescribeInstance EventNotification
	ec2: DescribeInstance EventWindows
	ec2: DescribeInstances
	ec2: Stato DescribeInstance
	ec2: Topologia DescribeInstance
	ec2: DescribeInstance TypeOfferings
	ec2: Tipi DescribeInstance

Prefisso del servizio	Azioni
	ec2: Gateway DescribeInternet
	ec2: Byosan DescribeIpam
	ec2: DescribeIpam Piscine
	ec2: DescribeIpam ResourceDiscoveries
	ec2: Associazioni DescribeIpam ResourceDiscovery
	ec2: DescribeIpams
	ec2: Ambiti DescribeIpam
	ec2:6 piscine DescribeIpv
	ec2: coppie DescribeKey
	ec2: Modelli DescribeLaunch
	ec2: DescribeLaunch TemplateVersions
	ec2: Tabelle DescribeLocal GatewayRoute
	ec2: Associazioni DescribeLocal GatewayRoute TableVirtual InterfaceGroup
	ec2: Associazioni DescribeLocal GatewayRoute TableVpc
	ec2: Gateway DescribeLocal
	ec2: DescribeLocal GatewayVirtual InterfaceGroups
	ec2: Interfacce DescribeLocal GatewayVirtual
	ec2: Istantanee DescribeLocked
	ec2: host DescribeMac
	ec2: DescribeManaged PrefixLists

Prefisso del servizio	Azioni
	ec2: Indirizzi DescribeMoving
	ec2: Gateway DescribeNat
	ec2: Acls DescribeNetwork
	ec2: DescribeNetwork InsightsAccess ScopeAnalyses
	ec2: Ambiti DescribeNetwork InsightsAccess
	ec2: DescribeNetwork InsightsAnalyses
	ec2: DescribeNetwork InsightsPaths
	ec2: DescribeNetwork InterfaceAttribute
	ec2: DescribeNetwork InterfacePermissions
	ec2: Interfacce DescribeNetwork
	ec2: Gruppi DescribePlacement
	ec2: Elenchi DescribePrefix
	ec2: DescribePrincipal IdFormat
	ec2: Pool Ipv4 DescribePublic
	ec2: DescribeRegions
	ec2: Attività DescribeReplace RootVolume
	ec2: Istanze DescribeReserved
	ec2: DescribeReserved InstancesListings
	ec2: DescribeReserved InstancesModifications
	ec2: DescribeReserved InstancesOfferings
	ec2: Tabelle DescribeRoute

Prefisso del servizio	Azioni
	ec2: DescribeScheduled InstanceAvailability
	ec2: Istanze DescribeScheduled
	ec2: DescribeSecurity GroupReferences
	ec2: DescribeSecurity GroupRules
	ec2: Gruppi DescribeSecurity
	ec2: Attributo DescribeSnapshot
	ec2: DescribeSnapshots
	ec2: DescribeSnapshot TierStatus
	ec2: DescribeSpot DatafeedSubscription
	ec2: DescribeSpot FleetInstances
	ec2: Storia DescribeSpot FleetRequest
	ec2: DescribeSpot FleetRequests
	ec2: DescribeSpot InstanceRequests
	ec2: DescribeSpot PriceHistory
	ec2: DescribeStale SecurityGroups
	ec2: DescribeStore ImageTasks
	ec2: DescribeSubnets
	ec2: DescribeTraffic MirrorFilters
	ec2: DescribeTraffic MirrorSessions
	ec2: DescribeTraffic MirrorTargets
	ec2: DescribeTransit GatewayAttachments

Prefisso del servizio	Azioni
	ec2: colleghi DescribeTransit GatewayConnect
	ec2: DescribeTransit GatewayConnects
	ec2: Domini DescribeTransit GatewayMulticast
	ec2: Allegati DescribeTransit GatewayPeering
	ec2: Tabelle DescribeTransit GatewayPolicy
	ec2: DescribeTransit GatewayRoute TableAnnouncements
	ec2: Tabelle DescribeTransit GatewayRoute
	ec2: Gateway DescribeTransit
	ec2: Allegati DescribeTransit GatewayVpc
	ec2: DescribeTrunk InterfaceAssociations
	ec2: DescribeVerified AccessEndpoints
	ec2: DescribeVerified AccessGroups
	ec2: DescribeVerified AccessInstance LoggingConfigurations
	ec2: DescribeVerified AccessInstances
	ec2: Fornitori DescribeVerified AccessTrust
	ec2: Attributo DescribeVolume
	ec2: DescribeVolumes
	ec2: Modifiche DescribeVolumes
	ec2: Stato DescribeVolume
	ec2: Attributo DescribeVpc
	ec2: DescribeVpc ClassicLink

Prefisso del servizio	Azioni
	ec2: DescribeVpc ClassicLink DnsSupport
	ec2: Notifiche DescribeVpc EndpointConnection
	ec2: DescribeVpc EndpointConnections
	ec2: punti finali DescribeVpc
	ec2: Configurazioni DescribeVpc EndpointService
	ec2: Autorizzazioni DescribeVpc EndpointService
	ec2: DescribeVpc EndpointServices
	ec2: DescribeVpc PeeringConnections
	ec2: DescribeVpcs
	ec2: Connessioni DescribeVpn
	ec2: Gateway DescribeVpn
	ec2: DetachClassic LinkVpc
	ec2: porta DetachInternet
	ec2: Interfaccia DetachNetwork
	ec2: Fornitore DetachVerified AccessTrust
	ec2: DetachVolume
	ec2: porta DetachVpn
	ec2: Trasferimento DisableAddress
	ec2: DisableAws NetworkPerformance MetricSubscription
	ec2: impostazione predefinita DisableEbs EncryptionBy
	ec2: Avvia DisableFast

Prefisso del servizio	Azioni
	ec2: DisableFast SnapshotRestores
	ec2: DisableImage
	ec2: Accesso DisableImage BlockPublic
	ec2: Deprecazione DisableImage
	ec2: DisableImage DeregistrationProtection
	ec2: Conto DisableIpam OrganizationAdmin
	ec2: DisableSerial ConsoleAccess
	ec2: Accesso DisableSnapshot BlockPublic
	ec2: DisableTransit GatewayRoute TablePropagation
	ec2: DisableVgw RoutePropagation
	ec2: DisableVpc ClassicLink
	ec2: DisableVpc ClassicLink DnsSupport
	ec2: DisassociateAddress
	ec2: Rete DisassociateClient VpnTarget
	ec2: Ruolo DisassociateEnclave CertificateIam
	ec2: DisassociateIam InstanceProfile
	ec2: DisassociateInstance EventWindow
	ec2: Byosan DisassociateIpam
	ec2: DisassociateIpam ResourceDiscovery
	ec2: DisassociateNat GatewayAddress
	ec2: Tabella DisassociateRoute

Prefisso del servizio	Azioni
	ec2: DisassociateSubnet CidrBlock
	ec2: Dominio DisassociateTransit GatewayMulticast
	ec2: Tabella DisassociateTransit GatewayPolicy
	ec2: Tabella DisassociateTransit GatewayRoute
	ec2: Interfaccia DisassociateTrunk
	ec2: DisassociateVpc CidrBlock
	ec2: Trasferimento EnableAddress
	ec2: EnableAws NetworkPerformance MetricSubscription
	ec2: impostazione predefinita EnableEbs EncryptionBy
	ec2: Avvia EnableFast
	ec2: EnableFast SnapshotRestores
	ec2: EnableImage
	ec2: Accesso EnableImage BlockPublic
	ec2: Deprecazione EnableImage
	ec2: EnableImage DeregistrationProtection
	ec2: Conto EnableIpam OrganizationAdmin
	ec2: Condivisione EnableReachability AnalyzerOrganization
	ec2: EnableSerial ConsoleAccess
	ec2: Accesso EnableSnapshot BlockPublic
	ec2: EnableTransit GatewayRoute TablePropagation
	ec2: EnableVgw RoutePropagation

Prefisso del servizio	Azioni
	ec2: IO EnableVolume
	ec2: EnableVpc ClassicLink
	ec2: EnableVpc ClassicLink DnsSupport
	ec2: Elenco ExportClient VpnClient CertificateRevocation
	ec2: Configurazione ExportClient VpnClient
	ec2: ExportImage
	ec2: ExportTransit GatewayRoutes
	ec2: GetAssociated EnclaveCertificate IamRoles
	ec2: IPv6 GetAssociated PoolCidrs
	ec2: Dati GetAws NetworkPerformance
	ec2: GetCapacity ReservationUsage
	ec2: GetCoip PoolUsage
	ec2: uscita GetConsole
	ec2: Immagine dello schermo GetConsole
	ec2: GetDefault CreditSpecification
	ec2: GetEbs DefaultKms KeyId
	ec2: impostazione predefinita GetEbs EncryptionBy
	ec2: modello GetFlow LogsIntegration
	ec2: Prenotazione GetGroups ForCapacity
	ec2: Anteprima GetHost ReservationPurchase
	ec2: GetImage BlockPublic AccessState

Prefisso del servizio	Azioni
	ec2: GetInstance MetadataDefaults
	ec2: Pub GetInstance TpmEk
	ec2: GetInstance TypesFrom InstanceRequirements
	ec2: GetInstance UefiData
	ec2: GetIpam AddressHistory
	ec2: GetIpam DiscoveredAccounts
	ec2: Indirizzi GetIpam DiscoveredPublic
	ec2: Sidri GetIpam DiscoveredResource
	ec2: GetIpam PoolAllocations
	ec2: GetIpam PoolCidrs
	ec2: GetIpam ResourceCidrs
	ec2: GetLaunch TemplateData
	ec2: Associazioni GetManaged PrefixList
	ec2: Iscrizioni GetManaged PrefixList
	ec2: Risultati GetNetwork InsightsAccess ScopeAnalysis
	ec2: GetNetwork InsightsAccess ScopeContent
	ec2: Dati GetPassword
	ec2: Citazione GetReserved InstancesExchange
	ec2: Vpc GetSecurity GroupsFor
	ec2: Stato GetSerial ConsoleAccess
	ec2: GetSnapshot BlockPublic AccessState

Prefisso del servizio	Azioni
	ec2: GetSpot PlacementScores
	ec2: GetSubnet CidrReservations
	ec2: Propagazioni GetTransit GatewayAttachment
	ec2: GetTransit GatewayMulticast DomainAssociations
	ec2: GetTransit GatewayPolicy TableAssociations
	ec2: GetTransit GatewayPolicy TableEntries
	ec2: GetTransit GatewayPrefix ListReferences
	ec2: GetTransit GatewayRoute TableAssociations
	ec2: GetTransit GatewayRoute TablePropagations
	ec2: Politica GetVerified AccessEndpoint
	ec2: Politica GetVerified AccessGroup
	ec2: GetVpn ConnectionDevice SampleConfiguration
	ec2: Tipi GetVpn ConnectionDevice
	ec2: Stato GetVpn TunnelReplacement
	ec2: Elenco ImportClient VpnClient CertificateRevocation
	ec2: ImportImage
	ec2: ImportInstance
	ec2: coppia ImportKey
	ec2: ImportSnapshot
	ec2: ImportVolume
	ec2: raccoglitore ListImages InRecycle

Prefisso del servizio	Azioni
	ec2: Raccoglitore ListSnapshots InRecycle
	ec2: LockSnapshot
	ec2: Attributo ModifyAddress
	ec2: ModifyAvailability ZoneGroup
	ec2: Prenotazione ModifyCapacity
	ec2: ModifyCapacity ReservationFleet
	ec2: ModifyClient VpnEndpoint
	ec2: ModifyDefault CreditSpecification
	ec2: ModifyEbs DefaultKms KeyId
	ec2: ModifyFleet
	ec2: ModifyFpga ImageAttribute
	ec2: ModifyHosts
	ec2: ModifyIdentity IdFormat
	ec2: Formato ModifyId
	ec2: Attributo ModifyImage
	ec2: Attributo ModifyInstance
	ec2: attributi ModifyInstance CapacityReservation
	ec2: ModifyInstance CreditSpecification
	ec2: Ora ModifyInstance EventStart
	ec2: ModifyInstance EventWindow
	ec2: ModifyInstance MaintenanceOptions

Prefisso del servizio	Azioni
	ec2: ModifyInstance MetadataDefaults
	ec2: ModifyInstance MetadataOptions
	ec2: Posizionamento ModifyInstance
	ec2: ModifyIpam
	ec2: Piscina ModifyIpam
	ec2: ModifyIpam ResourceCidr
	ec2: ModifyIpam ResourceDiscovery
	ec2: Ambito ModifyIpam
	ec2: Modello ModifyLaunch
	ec2: ModifyLocal GatewayRoute
	ec2: ModifyManaged PrefixList
	ec2: ModifyNetwork InterfaceAttribute
	ec2: Opzioni ModifyPrivate DnsName
	ec2: Istanze ModifyReserved
	ec2: ModifySecurity GroupRules
	ec2: Attributo ModifySnapshot
	ec2: livello ModifySnapshot
	ec2: ModifySpot FleetRequest
	ec2: Attributo ModifySubnet
	ec2: ModifyTraffic MirrorFilter NetworkServices
	ec2: Regola ModifyTraffic MirrorFilter

Prefisso del servizio	Azioni
	ec2: ModifyTraffic MirrorSession
	ec2: porta ModifyTransit
	ec2: ModifyTransit GatewayPrefix ListReference
	ec2: Allegato ModifyTransit GatewayVpc
	ec2: ModifyVerified AccessEndpoint
	ec2: Politica ModifyVerified AccessEndpoint
	ec2: ModifyVerified AccessGroup
	ec2: Politica ModifyVerified AccessGroup
	ec2: ModifyVerified AccessInstance
	ec2: ModifyVerified AccessInstance LoggingConfiguration
	ec2: Fornitore ModifyVerified AccessTrust
	ec2: ModifyVolume
	ec2: Attributo ModifyVolume
	ec2: Attributo ModifyVpc
	ec2: Punto finale ModifyVpc
	ec2: Notifica ModifyVpc EndpointConnection
	ec2: Configurazione ModifyVpc EndpointService
	ec2: ModifyVpc EndpointService PayerResponsibility
	ec2: Autorizzazioni ModifyVpc EndpointService
	ec2: Opzioni ModifyVpc PeeringConnection
	ec2: Locazione ModifyVpc

Prefisso del servizio	Azioni
	ec2: Connessione ModifyVpn
	ec2: ModifyVpn ConnectionOptions
	ec2: ModifyVpn TunnelCertificate
	ec2: ModifyVpn TunnelOptions
	ec2: MonitorInstances
	ec2: MoveAddress ToVpc
	ec2: Ipam MoveByoip CidrTo
	ec2: Sidro ProvisionByoip
	ec2: Byosan ProvisionIpam
	ec2: ProvisionIpam PoolCidr
	ec2: IPv4 ProvisionPublic PoolCidr
	ec2: Prenotazione PurchaseHost
	ec2: PurchaseReserved InstancesOffering
	ec2: Istanze PurchaseScheduled
	ec2: RebootInstances
	ec2: RegisterImage
	ec2: attributi RegisterInstance EventNotification
	ec2: RegisterTransit GatewayMulticast GroupMembers
	ec2: RegisterTransit GatewayMulticast GroupSources
	ec2: RejectTransit GatewayMulticast DomainAssociations
	ec2: Allegato RejectTransit GatewayPeering

Prefisso del servizio	Azioni
	ec2: Allegato RejectTransit GatewayVpc
	ec2: RejectVpc EndpointConnections
	ec2: RejectVpc PeeringConnection
	ec2: ReleaseAddress
	ec2: ReleaseHosts
	ec2: ReleaseIpam PoolAllocation
	ec2: Associazione ReplaceIam InstanceProfile
	ec2: ReplaceNetwork AclAssociation
	ec2: ReplaceNetwork AclEntry
	ec2: ReplaceRoute
	ec2: ReplaceRoute TableAssociation
	ec2: ReplaceTransit GatewayRoute
	ec2: Tunnel ReplaceVpn
	ec2: Stato ReportInstance
	ec2: Flotta RequestSpot
	ec2: Istanze RequestSpot
	ec2: Attributo ResetAddress
	ec2: ResetEbs DefaultKms KeyId
	ec2: ResetFpga ImageAttribute
	ec2: Attributo ResetImage
	ec2: Attributo ResetInstance

Prefisso del servizio	Azioni
	ec2: ResetNetwork InterfaceAttribute
	ec2: Attributo ResetSnapshot
	ec2: RestoreAddress ToClassic
	ec2: raccoglitore RestoreImage FromRecycle
	ec2: versione RestoreManaged PrefixList
	ec2: Bin RestoreSnapshot FromRecycle
	ec2: Livello RestoreSnapshot
	ec2: RevokeClient VpnIngress
	ec2: RevokeSecurity GroupEgress
	ec2: RevokeSecurity GroupIngress
	ec2: RunInstances
	ec2: Istanze RunScheduled
	ec2: SearchLocal GatewayRoutes
	ec2: Gruppi SearchTransit GatewayMulticast
	ec2: SearchTransit GatewayRoutes
	ec2: Interruzione SendDiagnostic
	ec2: StartInstances
	ec2: StartNetwork InsightsAccess ScopeAnalysis
	ec2: StartNetwork InsightsAnalysis
	ec2: verifica StartVpc EndpointService PrivateDns
	ec2: StopInstances

Prefisso del servizio	Azioni
	ec2: TerminateClient VpnConnections ec2: TerminateInstances ec2:6 indirizzi UnassignIpv ec2: UnassignPrivate IpAddresses ec2: Indirizzo UnassignPrivate NatGateway ec2: UnlockSnapshot ec2: UnmonitorInstances ec2: UpdateSecurity GroupRule DescriptionsEgress ec2: UpdateSecurity GroupRule DescriptionsIngress ec2: Sidro WithdrawByoip

Prefisso del servizio	Azioni
ecr	ecr: BatchCheck LayerAvailability ecr: Immagine BatchDelete ecr: Immagine BatchGet ecr: Configurazione BatchGet RepositoryScanning ecr: Carica CompleteLayer ecr: Regola CreatePull ThroughCache ecr: CreateRepository ecr: CreateRepository CreationTemplate ecr: Politica DeleteLifecycle ecr: Regola DeletePull ThroughCache ecr: Politica DeleteRegistry ecr: DeleteRepository ecr: DeleteRepository CreationTemplate ecr: Politica DeleteRepository ecr: DescribeImage ReplicationStatus ecr: DescribeImages ecr: DescribeImage ScanFindings ecr: Regole DescribePull ThroughCache ecr: DescribeRegistry ecr: DescribeRepositories ecr: gettone GetAuthorization

Prefisso del servizio	Azioni
	ecr: Livello GetDownload UrlFor ecr: Politica GetLifecycle ecr: GetLifecycle PolicyPreview ecr: Politica GetRegistry ecr: GetRegistry ScanningConfiguration ecr: Politica GetRepository ecr: Carica InitiateLayer ecr: ListImages ecr: PutImage ecr: PutImage ScanningConfiguration ecr: Politica PutRegistry ecr: PutRegistry ScanningConfiguration ecr: Configurazione PutReplication ecr: Scansione StartImage ecr: StartLifecycle PolicyPreview ecr: Regola UpdatePull ThroughCache ecr: Parte UploadLayer ecr: Regola ValidatePull ThroughCache

Prefisso del servizio	Azioni
ecr-public	ecr-public: BatchCheck LayerAvailability
	ecr-public: Immagine BatchDelete
	ecr-public: Carica CompleteLayer
	ecr-public: CreateRepository
	ecr-pubblico: DeleteRepository
	ecr-public: Politica DeleteRepository
	ecr-public: DescribeImages
	ecr-pubblico: DescribeRegistries
	ecr-pubblico: DescribeRepositories
	ecr-public: gettone GetAuthorization
	ecr-pubblico: GetRegistry CatalogData
	ecr-pubblico: GetRepository CatalogData
	ecr-public: Politica GetRepository
	ecr-public: Carica InitiateLayer
	ecr-public: PutImage
	ecr-pubblico: PutRegistry CatalogData
	ecr-pubblico: PutRepository CatalogData
	ecr-public: Politica SetRepository
	ecr-public: Parte UploadLayer

Prefisso del servizio	Azioni
ecs	ecs: Fornitore CreateCapacity ecs: CreateCluster ecs: CreateService ecs: set CreateTask ecs: Impostazioni DeleteAccount ecs: DeleteAttributes ecs: Fornitore DeleteCapacity ecs: DeleteCluster ecs: DeleteService ecs: definizioni DeleteTask ecs: Set DeleteTask ecs: Istanza DeregisterContainer ecs: definizione DeregisterTask ecs: fornitori DescribeCapacity ecs: DescribeClusters ecs: Istanze DescribeContainer ecs: DescribeServices ecs: definizione DescribeTask ecs: DescribeTasks ecs: set DescribeTask ecs: Punto finale DiscoverPoll

Prefisso del servizio	Azioni
	ecs: ExecuteCommand ecs: protezione GetTask ecs: Impostazioni ListAccount ecs: ListAttributes ecs: ListClusters ecs: Istanze ListContainer ecs: ListServices ecs: ListServices ByNamespace ecs: ListTask DefinitionFamilies ecs: definizioni ListTask ecs: ListTasks ecs: Impostazioni PutAccount ecs: PutAccount SettingDefault ecs: PutAttributes ecs: PutCluster CapacityProviders ecs: Istanza RegisterContainer ecs: definizione RegisterTask ecs: RunTask ecs: StartTask ecs: StopTask ecs: SubmitAttachment StateChanges

Prefisso del servizio	Azioni
	ecs: SubmitContainer StateChange ecs: SubmitTask StateChange ecs: Fornitore UpdateCapacity ecs: UpdateCluster ecs: Impostazioni UpdateCluster ecs: Agente UpdateContainer ecs: UpdateContainer InstancesState ecs: UpdateService ecs: set UpdateService PrimaryTask ecs: protezione UpdateTask ecs: Set UpdateTask

Prefisso del servizio	Azioni
eks	eks: Politica AssociateAccess
	es: AssociateEncryption Config
	es: AssociateIdentity ProviderConfig
	eks: CreateAccess Ingresso
	es: CreateAddon
	ex: CreateCluster
	ex: CreateEks AnywhereSubscription
	eks: CreateFargate Profilo
	eks: CreateNodegroup
	eks: DeleteAccess Ingresso
	es: DeleteAddon
	ex: DeleteCluster
	ex: DeleteEks AnywhereSubscription
	eks: DeleteFargate Profilo
	eks: DeleteNodegroup
	ex: DeletePod IdentityAssociation
	ex: DeregisterCluster
	eks: DescribeAccess Ingresso
	es: DescribeAddon
	eks: DescribeAddon Configurazione
	eks: DescribeAddon Versioni

Prefisso del servizio	Azioni
	es: DescribeCluster ex: DescribeEks AnywhereSubscription eks: DescribeFargate Profilo eks: DescribeIdentity ProviderConfig ex: DescribeInsight ex: DescribeNodegroup ex: DescribePod IdentityAssociation ex: DescribeUpdate eks: DisassociateAccess Politica es: DisassociateIdentity ProviderConfig eks: ListAccess Iscrizioni eks: ListAccess Politiche es: ListAddons ex: ListAssociated AccessPolicies ex: ListClusters ex: ListEks AnywhereSubscriptions eks: ListFargate Profili es: ListIdentity ProviderConfigs ex: ListInsights ex: ListNodegroups ex: ListPod IdentityAssociations

Prefisso del servizio	Azioni
	ex: ListUpdates ex: RegisterCluster eks: UpdateAccess Ingresso es: UpdateAddon es: UpdateCluster Config eks: versione UpdateCluster es: UpdateEks AnywhereSubscription es: UpdateNodegroup Config eks: versione UpdateNodegroup es: UpdatePod IdentityAssociation
elastic-inference	inferenza elastica: offerte DescribeAccelerator inferenza elastica: DescribeAccelerators inferenza elastica: tipi DescribeAccelerator

Prefisso del servizio	Azioni
elasticache	elasticache: Ingress AuthorizeCache SecurityGroup
	dolore elastico: BatchApply UpdateAction
	dolore elastico: BatchStop UpdateAction
	dolore elastico: CompleteMigration
	dolore elastico: CopyServerless CacheSnapshot
	dolore elastico: CopySnapshot
	elasticache: grappolo CreateCache
	dolore elastico: CreateCache ParameterGroup
	dolore elastico: CreateCache SecurityGroup
	dolore elastico: CreateCache SubnetGroup
	dolore elastico: CreateGlobal ReplicationGroup
	elasticache: Gruppo CreateReplication
	elasticache: Cache CreateServerless
	dolore elastico: CreateServerless CacheSnapshot
	dolore elastico: CreateSnapshot
	dolore elastico: CreateUser
	elasticache: Gruppo CreateUser
	elasticache: Gruppo DecreaseNode GroupsIn GlobalReplication
	elasticache: conta DecreaseReplica
	elasticache: Cluster DeleteCache
	dolore elastico: DeleteCache ParameterGroup

Prefisso del servizio	Azioni
	dolore elastico: DeleteCache SecurityGroup
	dolore elastico: DeleteCache SubnetGroup
	dolore elastico: DeleteGlobal ReplicationGroup
	elasticache: Gruppo DeleteReplication
	elasticache: Cache DeleteServerless
	dolore elastico: DeleteServerless CacheSnapshot
	dolore elastico: DeleteSnapshot
	dolore elastico: DeleteUser
	elasticache: Gruppo DeleteUser
	elasticache: grappoli DescribeCache
	elasticache: DescribeCache EngineVersions
	dolore elastico: DescribeCache ParameterGroups
	elasticache: Parametri DescribeCache
	elasticache: DescribeCache SecurityGroups
	dolore elastico: DescribeCache SubnetGroups
	dolore elastico: DescribeEngine DefaultParameters
	dolore elastico: DescribeEvents
	dolore elastico: DescribeGlobal ReplicationGroups
	elasticache: Gruppi DescribeReplication
	elasticache: DescribeReserved CacheNodes
	elasticache: Offerte DescribeReserved CacheNodes

Prefisso del servizio	Azioni
	elasticache: cache DescribeServerless
	elasticache: DescribeServerless CacheSnapshots
	elasticache: Aggiornamenti DescribeService
	elasticache: DescribeSnapshots
	elasticache: Azioni DescribeUpdate
	elasticache: Gruppi DescribeUser
	elasticache: DescribeUsers
	dolore elastico: DisassociateGlobal ReplicationGroup
	dolore elastico: ExportServerless CacheSnapshot
	dolore elastico: FailoverGlobal ReplicationGroup
	elasticache: Gruppo IncreaseNode GroupsIn GlobalReplication
	elasticache: conta IncreaseReplica
	elasticache: modifiche ListAllowed NodeType
	elasticache: Cluster ModifyCache
	dolore elastico: ModifyCache ParameterGroup
	dolore elastico: ModifyCache SubnetGroup
	dolore elastico: ModifyGlobal ReplicationGroup
	elasticache: Gruppo ModifyReplication
	elasticache: Configurazione ModifyReplication GroupShard
	elasticache: Cache ModifyServerless
	dolore elastico: ModifyUser

Prefisso del servizio	Azioni
	elasticache: Gruppo ModifyUser elasticache: Offerta PurchaseReserved CacheNodes elasticache: RebalanceSlots InGlobal ReplicationGroup elasticache: grappolo RebootCache dolore elastico: ResetCache ParameterGroup elasticache: Ingresso RevokeCache SecurityGroup dolore elastico: StartMigration dolore elastico: TestFailover dolore elastico: TestMigration

Prefisso del servizio	Azioni
elasticbeanstalk	elasticbeanstalk: Aggiornamento AbortEnvironment elasticbeanstalk: ApplyEnvironment ManagedAction gambo elastico di fagioli: AssociateEnvironment OperationsRole elasticbeanstalk:CheckDNSAvailability gambo elastico di fagioli: ComposeEnvironments gambo elastico di fagioli: CreateApplication elasticbeanstalk: versione CreateApplication elasticbeanstalk: Modello CreateConfiguration elasticbeanstalk: CreateEnvironment elasticbeanstalk: versione CreatePlatform elasticbeanstalk: Posizione CreateStorage elasticbeanstalk: DeleteApplication elasticbeanstalk: versione DeleteApplication elasticbeanstalk: Modello DeleteConfiguration elasticbeanstalk: Configurazione DeleteEnvironment elasticbeanstalk: Versione DeletePlatform elasticbeanstalk: DescribeAccount Attributi elasticbeanstalk: DescribeApplications elasticbeanstalk: Versioni DescribeApplication elasticbeanstalk: Opzioni DescribeConfiguration elasticbeanstalk: Impostazioni DescribeConfiguration

Prefisso del servizio	Azioni
	elasticbeanstalk: Health DescribeEnvironment elasticbeanstalk: Storia DescribeEnvironment ManagedAction elasticbeanstalk: DescribeEnvironment ManagedActions elasticbeanstalk: Risorse DescribeEnvironment elasticbeanstalk: DescribeEnvironments gambo elastico di fagioli: DescribeEvents elasticbeanstalk: Health DescribeInstances elasticbeanstalk: Versione DescribePlatform elasticbeanstalk: DisassociateEnvironment OperationsRole gambo elastico di fagioli: ListAvailable SolutionStacks gambo di fagioli elastico: rami ListPlatform elasticbeanstalk: Versioni ListPlatform elasticbeanstalk: RebuildEnvironment elasticbeanstalk: Informazioni RequestEnvironment elasticbeanstalk: Server RestartApp elasticbeanstalk: Informazioni RetrieveEnvironment elasticbeanstalk: CNames SwapEnvironment elasticbeanstalk: TerminateEnvironment gambo elastico di fagioli: UpdateApplication gambo elastico di fagioli: UpdateApplication ResourceLifecycle elasticbeanstalk: versione UpdateApplication

Prefisso del servizio	Azioni
	elasticbeanstalk: Modello UpdateConfiguration elasticbeanstalk: UpdateEnvironment elasticbeanstalk: Impostazioni ValidateConfiguration

Prefisso del servizio	Azioni
elasticfilesystem	elasticfilesystem: Point CreateAccess
	elasticfilesystem: CreateFile Sistema
	elasticfilesystem: Target CreateMount
	elasticfilesystem: Configurazione CreateReplication
	elasticfilesystem: Point DeleteAccess
	elasticfilesystem: DeleteFile Sistema
	elasticfilesystem: DeleteFile SystemPolicy
	elasticfilesystem: Target DeleteMount
	elasticfilesystem: Configurazione DeleteReplication
	elasticfilesystem: Punti DescribeAccess
	elasticfilesystem: Preferenze DescribeAccount
	elasticfilesystem: politica DescribeBackup
	elasticfilesystem: DescribeFile SystemPolicy
	elasticfilesystem: Sistemi DescribeFile
	elasticfilesystem: Configurazione DescribeLifecycle
	elasticfilesystem: obiettivi DescribeMount
	elasticfilesystem: Gruppi DescribeMount TargetSecurity
	elasticfilesystem: DescribeReplication Configurazioni
	elasticfilesystem: Gruppi ModifyMount TargetSecurity
	elasticfilesystem: Preferenze PutAccount
	elasticfilesystem: politica PutBackup

Prefisso del servizio	Azioni
	elasticfilesystem: PutFile SystemPolicy elasticfilesystem: Configurazione PutLifecycle elasticfilesystem: UpdateFile Sistema elasticfilesystem: UpdateFile SystemProtection

Prefisso del servizio	Azioni
elasticloadbalancing	elasticload balancing: certificati AddListener bilanciamento elastico del carico: AddTrust StoreRevocations bilanciamento elastico del carico: ApplySecurity GroupsTo LoadBalancer bilanciamento elastico del carico: sottoreti AttachLoad BalancerTo bilanciamento del carico elastico: verifica ConfigureHealth elasticloadbalancing: politica CreateApp CookieStickiness ElasticloadbalancingCookieStickiness: CreateLB Policy bilanciamento elastico del carico: CreateListener bilanciamento del carico elastico: Balancer CreateLoad bilanciamento del carico elastico: CreateLoad BalancerListeners bilanciamento elastico del carico: CreateLoad BalancerPolicy bilanciamento elastico del carico: CreateRule bilanciamento del carico elastico: gruppo CreateTarget bilanciamento del carico elastico: Store CreateTrust bilanciamento elastico del carico: DeleteListener bilanciamento del carico elastico: Balancer DeleteLoad bilanciamento del carico elastico: DeleteLoad BalancerListeners bilanciamento elastico del carico: DeleteLoad BalancerPolicy bilanciamento elastico del carico: DeleteRule bilanciamento del carico elastico: gruppo DeleteTarget

Prefisso del servizio	Azioni
	bilanciamento del carico elastico: Store DeleteTrust bilanciamento del carico elastico: DeregisterInstances FromLoad Balancer bilanciamento del carico elastico: DeregisterTargets bilanciamento elastico del carico: limiti DescribeAccount bilanciamento del carico elastico: Health DescribeInstance elasticloadbalancing: Certificati DescribeListener bilanciamento elastico del carico: DescribeListeners bilanciamento elastico del carico: DescribeLoad BalancerAttributes bilanciamento elastico del carico: DescribeLoad BalancerPolicies bilanciamento del carico elastico: tipi DescribeLoad BalancerPolicy bilanciamento del carico elastico: DescribeLoad bilanciatori bilanciamento del carico elastico: DescribeRules elasticloadbalancing:DescribeSSLPolicies bilanciamento elastico del carico: DescribeTarget GroupAttributes bilanciamento del carico elastico: gruppi DescribeTarget bilanciamento del carico elastico: Health DescribeTarget bilanciamento elastico del carico: DescribeTrust StoreAssociations bilanciamento elastico del carico: DescribeTrust StoreRevocations bilanciamento elastico del carico: Stores DescribeTrust bilanciamento elastico del carico: sottoreti DetachLoad BalancerF rom

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: DisableAvailability ZonesFor LoadBalancer bilanciamento elastico del carico: EnableAvailability ZonesFor LoadBalancer bilanciamento elastico del carico: GetTrust StoreCa Certifica tesBundle bilanciamento elastico del carico: contenuto GetTrust StoreRevo cation bilanciamento elastico del carico: ModifyListener bilanciamento elastico del carico: ModifyLoad BalancerAttributes bilanciamento elastico del carico: ModifyRule bilanciamento del carico elastico: gruppo ModifyTarget bilanciamento elastico del carico: ModifyTarget GroupAttributes bilanciamento elastico del carico: Store ModifyTrust bilanciamento del carico elastico: RegisterInstances WithLoad Balancer bilanciamento del carico elastico: RegisterTargets bilanciamento elastico del carico: certificati RemoveListener bilanciamento elastico del carico: RemoveTrust StoreRevocations bilanciamento elastico del carico: SetIp AddressType bilanciamento elastico del carico: certificato SSL SetLoad BalancerL istener elasticloadbalancing: Server SetLoad BalancerPolicies ForBackend

Prefisso del servizio	Azioni
	bilanciamento elastico del carico: SetLoad BalancerPolicies OfListener bilanciamento del carico elastico: priorità SetRule bilanciamento del carico elastico: gruppi SetSecurity bilanciamento elastico del carico: SetSubnets

Prefisso del servizio	Azioni
elastictranscoder	transcodificatore elastico: CancelJob
	transcodificatore elastico: CreateJob
	transcodificatore elastico: CreatePipeline
	transcodificatore elastico: CreatePreset
	transcodificatore elastico: DeletePipeline
	transcodificatore elastico: DeletePreset
	transcodificatore elastico: ListJobs ByPipeline
	transcodificatore elastico: ListJobs ByStatus
	transcodificatore elastico: ListPipelines
	transcodificatore elastico: ListPresets
	transcodificatore elastico: ReadJob
	transcodificatore elastico: ReadPipeline
	transcodificatore elastico: ReadPreset
	transcodificatore elastico: TestRole
	transcodificatore elastico: UpdatePipeline
	elastictranscoder: notifiche UpdatePipeline
	elastictranscoder: UpdatePipeline Stato

Prefisso del servizio	Azioni
emr-containers	emr-containers: CancelJob Esegui emr-containers: Modello CreateJob emr-containers: endpoint CreateManaged emr-containers: Configurazione CreateSecurity emr-containers: Cluster CreateVirtual emr-containers: Modello DeleteJob emr-containers: endpoint DeleteManaged emr-containers: Cluster DeleteVirtual emr-containers: Esegui DescribeJob emr-containers: Modello DescribeJob emr-containers: endpoint DescribeManaged emr-containers: Configurazione DescribeSecurity emr-containers: Cluster DescribeVirtual emr-containers: credenziali GetManaged EndpointSession emr-containers: esegue ListJob emr-containers: modelli ListJob emr-containers: endpoint ListManaged emr-containers: configurazioni ListSecurity emr-containers: cluster ListVirtual emr-containers: Esegui StartJob

Prefisso del servizio	Azioni
emr-serverless	emr-serverless: Esegui CancelJob
	emr-serverless: CreateApplication
	emr-senza server: DeleteApplication
	emr-senza server: GetApplication
	emr-serverless: Esegui GetDashboard ForJob
	emr-serverless: Esegui GetJob
	emr-serverless: ListApplications
	emr-serverless: viene eseguito ListJob
	emr-serverless: StartApplication
	emr-serverless: Esegui StartJob
	emr-serverless: StopApplication
	emr-senza server: UpdateApplication

Prefisso del servizio	Azioni
es	Sì: Connessione AcceptInbound Sì: AcceptInbound CrossCluster SearchConnection Sì: AssociatePackage Sì: AuthorizeVpc EndpointAccess es: CancelElasticsearch ServiceSoftware Aggiornamento Sì: CancelService SoftwareUpdate Sì: CreateDomain es: CreateElasticsearch Dominio es: CreateOutbound Connessione Sì: CreateOutbound CrossCluster SearchConnection Sì: CreatePackage Sì: CreateVpc Endpoint Sì: DeleteDomain es: DeleteElasticsearch Dominio Sì: DeleteElasticsearch ServiceRole es: DeleteInbound Connessione Sì: DeleteInbound CrossCluster SearchConnection es: DeleteOutbound Connessione Sì: DeleteOutbound CrossCluster SearchConnection Sì: DeletePackage Sì: DeleteVpc Endpoint

Prefisso del servizio	Azioni
	Sì: DescribeDomain Sì: DescribeDomain AutoTunes Sì: DescribeDomain ChangeProgress es: DescribeDomain Config es: DescribeDomain Health es: DescribeDomain nodi Sì: DescribeDomains Sì: DescribeDry RunProgress es: DescribeElasticsearch Dominio Sì: DescribeElasticsearch DomainConfig es: DescribeElasticsearch Domini es: DescribeElasticsearch InstanceType Limiti es: DescribeInbound Connessioni Sì: DescribeInbound CrossCluster SearchConnections Sì: DescribeInstance TypeLimits es: DescribeOutbound Connessioni Sì: DescribeOutbound CrossCluster SearchConnections Sì: DescribePackages es: DescribeReserved ElasticsearchInstance Offerte Sì: DescribeReserved ElasticsearchInstances Sì: DescribeReserved InstanceOfferings

Prefisso del servizio	Azioni
	es: DescribeReserved Istanze es: Endpoints DescribeVpc Sì: DissociatePackage Sì: GetCompatible ElasticsearchVersions es: GetCompatible Versioni es: GetData Fonte Sì: GetDomain MaintenanceStatus Sì: GetPackage VersionHistory es: GetUpgrade Storia es: GetUpgrade Stato es: ListData Fonti es: ListDomain nomi Sì: ListDomains ForPackage Sì: ListElasticsearch InstanceTypes es: ListElasticsearch Versioni Sì: ListInstance TypeDetails Sì: ListPackages ForDomain es: ListScheduled Azioni Sì: ListVersions Sì: ListVpc EndpointAccess es: ListVpc Endpoints

Prefisso del servizio	Azioni
	es: Dominio ListVpc EndpointsFor es: PurchaseReserved ElasticsearchInstance Offerta Sì: PurchaseReserved InstanceOffering es: RejectInbound Connessione Sì: RejectInbound CrossCluster SearchConnection Sì: RevokeVpc EndpointAccess es: StartDomain Manutenzione es: StartElasticsearch ServiceSoftware Aggiornamento Sì: StartService SoftwareUpdate es: UpdateData Fonte es: UpdateDomain Config Sì: UpdateElasticsearch DomainConfig Sì: UpdatePackage es: UpdateScheduled Azione es: UpdateVpc Endpoint Sì: UpgradeDomain es: UpgradeElasticsearch Dominio

Prefisso del servizio	Azioni
events	eventi: ActivateEvent fonte
	eventi: CancelReplay
	eventi: CreateApi Destinazione
	eventi: CreateArchive
	eventi: CreateConnection
	eventi: CreateEndpoint
	eventi: CreateEvent Bus
	eventi: CreatePartner EventSource
	eventi: DeactivateEvent fonte
	eventi: DeauthorizeConnection
	eventi: DeleteApi Destinazione
	eventi: DeleteArchive
	eventi: DeleteConnection
	eventi: DeleteEndpoint
	eventi: DeleteEvent Bus
	eventi: DeletePartner EventSource
	eventi: DeleteRule
	eventi: DescribeApi Destinazione
	eventi: DescribeArchive
	eventi: DescribeConnection
	eventi: DescribeEndpoint

Prefisso del servizio	Azioni
	eventi: DescribeEvent Bus
	eventi: DescribeEvent Fonte
	eventi: DescribePartner EventSource
	eventi: DescribeReplay
	eventi: DescribeRule
	eventi: DisableRule
	eventi: EnableRule
	eventi: ListApi Destinazioni
	eventi: ListArchives
	eventi: ListConnections
	eventi: ListEndpoints
	eventi: ListEvent Autobus
	eventi: ListEvent Fonti
	eventi: ListPartner EventSource conti
	eventi: ListPartner EventSources
	eventi: ListReplays
	eventi: ListRule NamesBy Target
	eventi: ListRules
	eventi: ListTargets ByRule
	eventi: PutPermission
	eventi: PutRule

Prefisso del servizio	Azioni
	eventi: PutTargets
	eventi: RemovePermission
	eventi: RemoveTargets
	eventi: StartReplay
	eventi: TestEvent Pattern
	eventi: UpdateApi Destinazione
	eventi: UpdateArchive
	eventi: UpdateConnection
	eventi: UpdateEndpoint

Prefisso del servizio	Azioni
evidently	evidentemente: CreateExperiment evidentemente: CreateFeature evidentemente: CreateLaunch evidentemente: CreateProject evidentemente: CreateSegment evidentemente: DeleteExperiment evidentemente: DeleteFeature evidentemente: DeleteLaunch evidentemente: DeleteProject evidentemente: DeleteSegment evidentemente: GetExperiment evidentemente: risultati GetExperiment evidentemente: GetFeature evidentemente: GetLaunch evidentemente: GetProject evidentemente: GetSegment evidentemente: ListExperiments evidentemente: ListFeatures evidentemente: ListLaunches evidentemente: ListProjects evidentemente: riferimenti ListSegment

Prefisso del servizio	Azioni
	evidentemente: ListSegments evidentemente: StartExperiment evidentemente: StartLaunch evidentemente: StopExperiment evidentemente: StopLaunch evidentemente: modello TestSegment evidentemente: UpdateExperiment evidentemente: UpdateFeature evidentemente: UpdateLaunch evidentemente: UpdateProject evidentemente: UpdateProject DataDelivery

Prefisso del servizio	Azioni
finspace	spazio interno: CreateEnvironment finspace: set di modifiche CreateKx finspace: Cluster CreateKx finspace: banca dati CreateKx finspace: Visualizzazione dati CreateKx finspace: Ambiente CreateKx finspace: CreateKx ScalingGroup finspace: Utente CreateKx finspace: Volume CreateKx spazio interno: CreateUser spazio interno: DeleteEnvironment finspace: Cluster DeleteKx finspace: DeleteKx ClusterNode finspace: banca dati DeleteKx finspace: Visualizzazione dati DeleteKx finspace: Ambiente DeleteKx finspace: DeleteKx ScalingGroup finspace: Utente DeleteKx finspace: Volume DeleteKx spazio interno: GetEnvironment finspace: set di modifiche GetKx

Prefisso del servizio	Azioni
	finspace: Cluster GetKx
	finspace: GetKx ConnectionString
	finspace: banca dati GetKx
	finspace: Visualizzazione dati GetKx
	finspace: Ambiente GetKx
	finspace: GetKx ScalingGroup
	finspace: Utente GetKx
	finspace: Volume GetKx
	finspace: Stato GetLoad SampleData SetGroup IntoEnvironment
	finspace: GetUser
	spazio interno: ListEnvironments
	finspace: set di modifiche ListKx
	finspace: ListKx ClusterNodes
	finspace: cluster ListKx
	finspace: database ListKx
	finspace: Visualizzazioni dati ListKx
	finspaceListKx: Ambienti
	finspace: ListKx ScalingGroups
	finspace: Utenti ListKx
	finspace: Volumi ListKx
	finspace: ListUsers

Prefisso del servizio	Azioni
	finspace: Ambiente LoadSample DataSet GroupInto finspace: Parola d'ordine ResetUser finspace: UpdateEnvironment finspace: Configurazione UpdateKx ClusterCode finspace: UpdateKx ClusterDatabases finspace: banca dati UpdateKx finspace: Visualizzazione dati UpdateKx finspace: Ambiente UpdateKx finspace: UpdateKx EnvironmentNetwork finspace: Utente UpdateKx finspace: Volume UpdateKx spazio interno: UpdateUser
firehose	firehose: Stream CreateDelivery firehose: Stream DeleteDelivery firehose: Stream DescribeDelivery firehose: Streams ListDelivery manichetta antincendio: StartDelivery StreamEncryption manichetta antincendio: StopDelivery StreamEncryption manichetta antincendio: UpdateDestination

Prefisso del servizio	Azioni
fis	fis: modello CreateExperiment fis: CreateTarget AccountConfiguration fis: modello DeleteExperiment fis: DeleteTarget AccountConfiguration pesce: GetAction pesce: GetExperiment fis: configurazione GetExperiment TargetAccount fis: Modello GetExperiment fis: GetTarget AccountConfiguration pesce: GetTarget ResourceType pesce: ListActions pesce: ListExperiment ResolvedTargets pesce: ListExperiments fis: configurazioni ListExperiment TargetAccount fis: Modelli ListExperiment fis: ListTarget AccountConfigurations pesce: ListTarget ResourceTypes pesce: StartExperiment pesce: StopExperiment fis: modello UpdateExperiment fis: UpdateTarget AccountConfiguration

Prefisso del servizio	Azioni
fms	fms: Conto AssociateAdmin fms: AssociateThird PartyFirewall fms: Risorsa BatchAssociate fms: Risorsa BatchDisassociate fms: Elenco DeleteApps fms: Canale DeleteNotification fms: DeletePolicy fms: Elenco DeleteProtocols fms: Imposta DeleteResource fms: Conto DisassociateAdmin fms: DisassociateThird PartyFirewall fms: Conto GetAdmin fms: Ambito GetAdmin fms: Elenco GetApps fms: Dettaglio GetCompliance fms: Canale GetNotification fms: GetPolicy fms: stato GetProtection fms: Elenco GetProtocols fms: Imposta GetResource fms: GetThird PartyFirewall AssociationStatus

Prefisso del servizio	Azioni
	fms: Dettagli GetViolation fms: Organizzazione ListAdmin AccountsFor fms: ListAdmins ManagingAccount fms: elenchi ListApps fms: stato ListCompliance fms: Risorse ListDiscovered fms: conti ListMember fms: ListPolicies fms: elenchi ListProtocols fms: ListResource SetResources fms: set ListResource fms: ListThird PartyFirewall FirewallPolicies fms: Conto PutAdmin fms: Elenco PutApps fms: Canale PutNotification fms: PutPolicy fms: Elenco PutProtocols fms: Imposta PutResource

Prefisso del servizio	Azioni
frauddetector	rilevatore di frodi: variabile BatchCreate
	rilevatore di frodi: variabile BatchGet
	rilevatore di frodi: CancelBatch ImportJob
	rilevatore di frodi: CancelBatch PredictionJob
	rilevatore di frodi: CreateBatch ImportJob
	rilevatore di frodi: CreateBatch PredictionJob
	frauddetector: versione CreateDetector
	rilevatore di frodi: CreateList
	rilevatore di frodi: CreateModel
	frauddetector: versione CreateModel
	rilevatore di frodi: CreateRule
	rilevatore di frodi: CreateVariable
	rilevatore di frodi: DeleteBatch ImportJob
	rilevatore di frodi: DeleteBatch PredictionJob
	rilevatore di frodi: DeleteDetector
	frauddetector: versione DeleteDetector
	frauddetector: Tipo DeleteEntity
	rilevatore di frodi: DeleteEvent
	rilevatore di frodi: tipo DeleteEvents ByEvent
	rilevatore di frodi: Tipo DeleteEvent
	frauddetector: Modello DeleteExternal

Prefisso del servizio	Azioni
	rilevatore di frodi: DeleteLabel
	rilevatore di frodi: DeleteList
	rilevatore di frodi: DeleteModel
	frauddetector: versione DeleteModel
	rilevatore di frodi: DeleteOutcome
	rilevatore di frodi: DeleteRule
	rilevatore di frodi: DeleteVariable
	rilevatore di frodi: DescribeDetector
	rilevatore di frodi: versioni DescribeModel
	rilevatore di frodi: GetBatch ImportJobs
	rilevatore di frodi: GetBatch PredictionJobs
	rilevatore di frodi: stato GetDelete EventsBy EventType
	rilevatore di frodi: GetDetectors
	frauddetector: versione GetDetector
	frauddetector: tipi GetEntity
	rilevatore di frodi: GetEvent
	frauddetector: previsione GetEvent
	rilevatore di frodi: GetEvent PredictionMetadata
	rilevatore di frodi: tipi GetEvent
	frauddetector: Modelli GetExternal
	Rilevatore di frodi: getKMS EncryptionKey

Prefisso del servizio	Azioni
	rilevatore di frodi: GetLabels
	rilevatore di frodi: Elementi GetList
	frauddetector: Metadati GetLists
	rilevatore di frodi: GetModels
	frauddetector: versione GetModel
	rilevatore di frodi: GetOutcomes
	rilevatore di frodi: GetRules
	rilevatore di frodi: GetVariables
	frauddetector: previsioni ListEvent
	rilevatore di frodi: PutDetector
	rilevatore di frodi: tipo PutEntity
	rilevatore di frodi: Tipo PutEvent
	frauddetector: Modello PutExternal
	Rilevatore di frodi: putkms EncryptionKey
	rilevatore di frodi: PutLabel
	rilevatore di frodi: PutOutcome
	rilevatore di frodi: SendEvent
	frauddetector: versione UpdateDetector
	rilevatore di frodi: UpdateDetector VersionMetadata
	rilevatore di frodi: UpdateDetector VersionStatus
	rilevatore di frodi: etichetta UpdateEvent

Prefisso del servizio	Azioni
	rilevatore di frodi: UpdateList rilevatore di frodi: UpdateModel frauddetector: versione UpdateModel rilevatore di frodi: UpdateModel VersionStatus frauddetector: metadati UpdateRule frauddetector: versione UpdateRule rilevatore di frodi: UpdateVariable

Prefisso del servizio	Azioni
fsx	fax: AssociateFile SystemAliases fax: CancelData RepositoryTask fax: CopyBackup fax: CreateData RepositoryTask fsx: cache CreateFile fsx: Sistema CreateFile fsx: Backup CreateFile SystemFrom fsx: CreateSnapshot fax: CreateStorage VirtualMachine fax: CreateVolume fax: CreateVolume FromBackup fax: DeleteBackup fsx: cache DeleteFile fsx: Sistema DeleteFile fsx: DeleteSnapshot fax: DeleteStorage VirtualMachine fax: DeleteVolume fax: DescribeBackups fax: DescribeData RepositoryAssociations fax: DescribeData RepositoryTasks fsx: cache DescribeFile

Prefisso del servizio	Azioni
	fsx: DescribeFile SystemAliases fsx: Sistemi DescribeFile fsx: DescribeShared VpcConfiguration fax: DescribeSnapshots fax: DescribeStorage VirtualMachines fax: DescribeVolumes fax: DisassociateFile SystemAliases fsx: Blocchi V3 ReleaseFile SystemNfs fax: RestoreVolume FromSnapshot fax: StartMisconfigured StateRecovery fax: UpdateData RepositoryAssociation fsx: cache UpdateFile fsx: Sistema UpdateFile fsx: UpdateShared VpcConfiguration fax: UpdateSnapshot fax: UpdateStorage VirtualMachine fax: UpdateVolume

Prefisso del servizio	Azioni
gamelift	rinnovamento del gioco: AcceptMatch
	gamelift: server ClaimGame
	rinnovamento del gioco: CreateAlias
	rinnovamento del gioco: CreateBuild
	rinnovamento del gioco: CreateContainer GroupDefinition
	rinnovamento del gioco: CreateFleet
	gamelift: sedi CreateFleet
	gamelift: CreateGame ServerGroup
	gamelift: sessione CreateGame
	rinnovamento del gioco: CreateGame SessionQueue
	rinnovamento del gioco: CreateLocation
	gamelift: Configurazione CreateMatchmaking
	gamelift: CreateMatchmaking RuleSet
	gamelift: sessione CreatePlayer
	gamelift: Sessioni CreatePlayer
	gamelift: CreateScript
	rinnovamento del gioco: CreateVpc PeeringAuthorization
	rinnovamento del gioco: CreateVpc PeeringConnection
	rinnovamento del gioco: DeleteAlias
	rinnovamento del gioco: DeleteBuild
	rinnovamento del gioco: DeleteContainer GroupDefinition

Prefisso del servizio	Azioni
	rinnovamento del gioco: DeleteFleet
	gamelift: sedi DeleteFleet
	gamelift: DeleteGame ServerGroup
	rinnovamento del gioco: DeleteGame SessionQueue
	rinnovamento del gioco: DeleteLocation
	gamelift: Configurazione DeleteMatchmaking
	gamelift: DeleteMatchmaking RuleSet
	gamelift: Politica DeleteScaling
	gamelift: DeleteScript
	rinnovamento del gioco: DeleteVpc PeeringAuthorization
	rinnovamento del gioco: DeleteVpc PeeringConnection
	rinnovamento del gioco: DeregisterCompute
	gamelift: server DeregisterGame
	rinnovamento del gioco: DescribeAlias
	rinnovamento del gioco: DescribeBuild
	rinnovamento del gioco: DescribeCompute
	rinnovamento del gioco: DescribeContainer GroupDefinition
	gamelift: descrivi EC2 InstanceLimits
	gamelift: Attributi DescribeFleet
	gamelift: Capacità DescribeFleet
	gamelift: Eventi DescribeFleet

Prefisso del servizio	Azioni
	gamelift: DescribeFleet LocationAttributes
	rinnovamento del gioco: DescribeFleet LocationCapacity
	rinnovamento del gioco: DescribeFleet LocationUtilization
	rinnovamento del gioco: DescribeFleet PortSettings
	gamelift: Utilizzo DescribeFleet
	gamelift: server DescribeGame
	rinnovamento del gioco: DescribeGame ServerGroup
	rinnovamento del gioco: DescribeGame ServerInstances
	rinnovamento del gioco: DescribeGame SessionDetails
	rinnovamento del gioco: DescribeGame SessionPlacement
	rinnovamento del gioco: DescribeGame SessionQueues
	gamelift: sessioni DescribeGame
	gamelift: DescribeInstances
	rinnovamento del gioco: DescribeMatchmaking
	gamelift: configurazioni DescribeMatchmaking
	gamelift: DescribeMatchmaking RuleSets
	gamelift: sessioni DescribePlayer
	gamelift: Configurazione DescribeRuntime
	gamelift: Politiche DescribeScaling
	gamelift: DescribeScript
	rinnovamento del gioco: DescribeVpc PeeringAuthorizations

Prefisso del servizio	Azioni
	rinnovamento del gioco: DescribeVpc PeeringConnections
	gamelift: Accesso GetCompute
	gamelift: GetCompute AuthToken
	gamelift: URL GetGame SessionLog
	gamelift: Accesso GetInstance
	gamelift: ListAliases
	rinnovamento del gioco: ListBuilds
	rinnovamento del gioco: ListCompute
	rinnovamento del gioco: ListContainer GroupDefinitions
	rinnovamento del gioco: ListFleets
	rinnovamento del gioco: ListGame ServerGroups
	gamelift: server ListGame
	gamelift: ListLocations
	rinnovamento del gioco: ListScripts
	gamelift: Politica PutScaling
	gamelift: RegisterCompute
	gamelift: server RegisterGame
	gamelift: Credenziali RequestUpload
	gamelift: ResolveAlias
	rinnovamento del gioco: ResumeGame ServerGroup
	gamelift: sessioni SearchGame

Prefisso del servizio	Azioni
	gamelift: Azioni StartFleet
	gamelift: StartGame SessionPlacement
	gamelift: Backfill StartMatch
	rinnovamento del gioco: StartMatchmaking
	gamelift: azioni StopFleet
	gamelift: StopGame SessionPlacement
	rinnovamento del gioco: StopMatchmaking
	rinnovamento del gioco: SuspendGame ServerGroup
	rinnovamento del gioco: UpdateAlias
	rinnovamento del gioco: UpdateBuild
	gamelift: Attributi UpdateFleet
	gamelift: Capacità UpdateFleet
	gamelift: UpdateFleet PortSettings
	gamelift: server UpdateGame
	rinnovamento del gioco: UpdateGame ServerGroup
	gamelift: sessione UpdateGame
	rinnovamento del gioco: UpdateGame SessionQueue
	gamelift: Configurazione UpdateMatchmaking
	gamelift: Configurazione UpdateRuntime
	gamelift: UpdateScript
	rinnovamento del gioco: ValidateMatchmaking RuleSet

Prefisso del servizio	Azioni
geo	geo: Consumatore AssociateTracker geo: Storia BatchDelete DevicePosition geo: Geofence BatchDelete geo: Geofences BatchEvaluate geo: BatchGet DevicePosition geo: Geofence BatchPut geo: BatchUpdate DevicePosition geo: CalculateRoute geo: Matrix CalculateRoute geo: Collezione CreateGeofence geo: CreateMap geo: Indice CreatePlace geo: Calcolatrice CreateRoute geo: CreateTracker geo: Collezione DeleteGeofence geo: DeleteKey geo: DeleteMap geo: Indice DeletePlace geo: Calcolatrice DeleteRoute geo: DeleteTracker geo: Collezione DescribeGeofence

Prefisso del servizio	Azioni
	geo: DescribeKey geo: DescribeMap geo: Indice DescribePlace geo: Calcolatrice DescribeRoute geo: DescribeTracker geo: Consumatore DisassociateTracker geo: Posizione GetDevice geo: GetDevice PositionHistory geo: GetGeofence geo: Glifi GetMap geo: Sprites GetMap geo: GetMap StyleDescriptor geo: Tile GetMap geo: GetPlace geo: Posizioni ListDevice geo: Collezioni ListGeofence geo: ListGeofences geo: ListKeys geo: ListMaps geo: Indici ListPlace geo: Calcolatrici ListRoute

Prefisso del servizio	Azioni
	geo: Consumatori ListTracker geo: ListTrackers geo: PutGeofence geo: Posizione SearchPlace IndexFor geo: Suggerimenti SearchPlace IndexFor geo: Testo SearchPlace IndexFor geo: Collezione UpdateGeofence geo: UpdateKey geo: UpdateMap geo: Indice UpdatePlace geo: Calcolatrice UpdateRoute geo: UpdateTracker

Prefisso del servizio	Azioni
glacier	glacier: Carica AbortMultipart
	glacier: Serratura AbortVault
	glacier: Carica CompleteMultipart
	glacier: Serratura CompleteVault
	ghiacciaio: CreateVault
	ghiacciaio: DeleteArchive
	ghiacciaio: DeleteVault
	ghiacciaio: DeleteVault AccessPolicy
	glacier: notifiche DeleteVault
	ghiacciaio: DescribeJob
	ghiacciaio: DescribeVault
	ghiacciaio: GetData RetrievalPolicy
	ghiacciaio: uscita GetJob
	ghiacciaio: GetVault AccessPolicy
	ghiacciaio: serratura GetVault
	glacier: notifiche GetVault
	ghiacciaio: InitiateJob
	glacier: Carica InitiateMultipart
	glacier: Serratura InitiateVault
	ghiacciaio: ListJobs
	glacier: Caricamenti ListMultipart

Prefisso del servizio	Azioni
	ghiacciaio: ListParts ghiacciaio: capacità ListProvisioned ghiacciaio: ListVaults ghiacciaio: capacità PurchaseProvisioned ghiacciaio: SetData RetrievalPolicy ghiacciaio: SetVault AccessPolicy glacier: notifiche SetVault ghiacciaio: UploadArchive ghiacciaio: Parte UploadMultipart

Prefisso del servizio	Azioni
grafana	grafana: AssociateLicense grafana: CreateWorkspace grafana: CreateWorkspace ApiKey grafana: DeleteWorkspace grafana: DeleteWorkspace ApiKey grafana: DescribeWorkspace grafana: autenticazione DescribeWorkspace grafana: Configurazione DescribeWorkspace grafana: DisassociateLicense grafana: ListPermissions grafana: ListVersions grafana: ListWorkspaces grafana: UpdatePermissions grafana: UpdateWorkspace grafana: autenticazione UpdateWorkspace grafana: Configurazione UpdateWorkspace

Prefisso del servizio	Azioni
greengrass	erba verde: AssociateRole ToGroup
	greengrass: Conto AssociateService RoleTo
	greengrass: Dispositivo BatchAssociate ClientDevice WithCore
	greengrass: Dispositivo BatchDisassociate ClientDevice FromCore
	erba verde: CancelDeployment
	greengrass: versione CreateComponent
	greengrass: Definizione CreateConnector
	erba verde: CreateConnector DefinitionVersion
	greengrass: definizione CreateCore
	erba verde: CreateCore DefinitionVersion
	erba verde: CreateDeployment
	greengrass: definizione CreateDevice
	erba verde: CreateDevice DefinitionVersion
	greengrass: definizione CreateFunction
	erba verde: CreateFunction DefinitionVersion
	erba verde: CreateGroup
	erba verde: CreateGroup CertificateAuthority
	greengrass: versione CreateGroup
	greengrass: Definizione CreateLogger
	erba verde: CreateLogger DefinitionVersion
	greengrass: definizione CreateResource

Prefisso del servizio	Azioni
	erba verde: CreateResource DefinitionVersion
	erba verde: CreateSoftware UpdateJob
	greengrass: definizione CreateSubscription
	erba verde: CreateSubscription DefinitionVersion
	erba verde: DeleteComponent
	greengrass: definizione DeleteConnector
	greengrass: definizione DeleteCore
	greengrass: Dispositivo DeleteCore
	erba verde: DeleteDeployment
	greengrass: definizione DeleteDevice
	greengrass: definizione DeleteFunction
	erba verde: DeleteGroup
	greengrass: definizione DeleteLogger
	greengrass: definizione DeleteResource
	greengrass: definizione DeleteSubscription
	erba verde: DescribeComponent
	erba verde: DisassociateRole FromGroup
	greengrass: Conto DisassociateService RoleFrom
	greengrass: Ruolo GetAssociated
	greengrass: GetBulk DeploymentStatus
	erba verde: GetComponent

Prefisso del servizio	Azioni
	erba verde: GetComponent VersionArtifact
	greengrass: Informazioni GetConnectivity
	greengrass: Definizione GetConnector
	erba verde: GetConnector DefinitionVersion
	greengrass: definizione GetCore
	erba verde: GetCore DefinitionVersion
	greengrass: Dispositivo GetCore
	erba verde: GetDeployment
	greengrass: Stato GetDeployment
	greengrass: definizione GetDevice
	erba verde: GetDevice DefinitionVersion
	greengrass: definizione GetFunction
	erba verde: GetFunction DefinitionVersion
	erba verde: GetGroup
	erba verde: GetGroup CertificateAuthority
	erba verde: GetGroup CertificateConfiguration
	greengrass: versione GetGroup
	greengrass: Definizione GetLogger
	erba verde: GetLogger DefinitionVersion
	greengrass: definizione GetResource
	erba verde: GetResource DefinitionVersion

Prefisso del servizio	Azioni
	greengrass: Conto GetService RoleFor
	greengrass: definizione GetSubscription
	erba verde: GetSubscription DefinitionVersion
	erba verde: GetThing RuntimeConfiguration
	greengrass: Rapporti ListBulk DeploymentDetailed
	greengrass: Implementazioni ListBulk
	greengrass: Dispositivo ListClient DevicesAssociated WithCore
	erba verde: ListComponents
	greengrass: Versioni ListComponent
	greengrass: Definizioni ListConnector
	erba verde: ListConnector DefinitionVersions
	greengrass: definizioni ListCore
	erba verde: ListCore DefinitionVersions
	greengrass: Dispositivi ListCore
	greengrass: ListDeployments
	greengrass: definizioni ListDevice
	erba verde: ListDevice DefinitionVersions
	greengrass: Implementazioni ListEffective
	greengrass: definizioni ListFunction
	erba verde: ListFunction DefinitionVersions
	erba verde: ListGroup CertificateAuthorities

Prefisso del servizio	Azioni
	erba verde: ListGroups
	greengrass: Versioni ListGroup
	greengrass: Componenti ListInstalled
	greengrass: Definizioni ListLogger
	erba verde: ListLogger DefinitionVersions
	greengrass: definizioni ListResource
	erba verde: ListResource DefinitionVersions
	greengrass: definizioni ListSubscription
	erba verde: ListSubscription DefinitionVersions
	erba verde: ResetDeployments
	greengrass: Implementazione StartBulk
	greengrass: Implementazione StopBulk
	greengrass: Informazioni UpdateConnectivity
	greengrass: Definizione UpdateConnector
	greengrass: definizione UpdateCore
	greengrass: definizione UpdateDevice
	greengrass: definizione UpdateFunction
	erba verde: UpdateGroup
	erba verde: UpdateGroup CertificateConfiguration
	greengrass: definizione UpdateLogger
	greengrass: definizione UpdateResource

Prefisso del servizio	Azioni
	greengrass: definizione UpdateSubscription erba verde: UpdateThing RuntimeConfiguration

Prefisso del servizio	Azioni
groundstation	stazione a terra: CancelContact stazione a terra: CreateConfig stazione a terra: CreateDataflow EndpointGroup stazione a terra: CreateEphemeris groundstation: Profilo CreateMission stazione di terra: DeleteConfig stazione a terra: DeleteDataflow EndpointGroup stazione a terra: DeleteEphemeris groundstation: Profilo DeleteMission stazione di terra: DescribeContact stazione a terra: DescribeEphemeris stazione a terra: GetConfig stazione a terra: GetDataflow EndpointGroup groundstation: Utilizzo GetMinute groundstation: Profilo GetMission stazione di terra: GetSatellite stazione a terra: ListConfigs stazione a terra: ListContacts stazione a terra: ListDataflow EndpointGroups stazione a terra: ListEphemerides stazione di terra: stazioni ListGround

Prefisso del servizio	Azioni
	groundstation: Profili ListMission stazione di terra: ListSatellites stazione a terra: RegisterAgent stazione a terra: ReserveContact stazione di terra: stato UpdateAgent stazione a terra: UpdateConfig stazione a terra: UpdateEphemeris groundstation: Profilo UpdateMission

Prefisso del servizio	Azioni
guardduty	guardduty: Invito AcceptAdministrator
	guardduty: AcceptInvitation
	servizio di guardia: ArchiveFindings
	servizio di guardia: CreateDetector
	servizio di guardia: CreateFilter
	guardduty:CreateIPSet
	servizio di guardia: CreateMembers
	guardduty: Destinazione CreatePublishing
	guardduty: risultati CreateSample
	servizio di guardia: CreateThreat IntelSet
	servizio di guardia: DeclineInvitations
	servizio di guardia: DeleteDetector
	servizio di guardia: DeleteFilter
	servizio di guardia: DeleteInvitations
	guardduty:DeleteIPSet
	servizio di guardia: DeleteMembers
	guardduty: Destinazione DeletePublishing
	servizio di guardia: DeleteThreat IntelSet
	guardduty: scansioni DescribeMalware
	guardduty: Configurazione DescribeOrganization
	guardduty: Destinazione DescribePublishing

Prefisso del servizio	Azioni
	servizio di guardia: DisableOrganization AdminAccount
	servizio di guardia: DisassociateFrom AdministratorAccount
	servizio di guardia: DisassociateFrom MasterAccount
	servizio di guardia: DisassociateMembers
	servizio di guardia: EnableOrganization AdminAccount
	guardduty: Conto GetAdministrator
	guardduty: Statistiche GetCoverage
	guardduty: GetDetector
	servizio di guardia: GetFilter
	servizio di guardia: GetFindings
	guardduty: Statistiche GetFindings
	guardduty: Conta GetInvitations
	guardduty:GetIPSet
	servizio di guardia: GetMalware ScanSettings
	guardduty: Conto GetMaster
	guardduty: Rilevatori GetMember
	guardduty: GetMembers
	guardduty: Statistiche GetOrganization
	guardduty: giorni GetRemaining FreeTrial
	servizio di guardia: GetThreat IntelSet
	guardduty: Statistiche GetUsage

Prefisso del servizio	Azioni
	guardduty: InviteMembers
	servizio di guardia: ListCoverage
	servizio di guardia: ListDetectors
	servizio di guardia: ListFilters
	servizio di guardia: ListFindings
	servizio di guardia: ListInvitations
	guardduty:ListIPSets
	servizio di guardia: ListMembers
	servizio di guardia: ListOrganization AdminAccounts
	guardduty: Destinazioni ListPublishing
	servizio di guardia: ListThreat IntelSets
	guardduty: telemetria SendSecurity
	guarddutyStartMalware: Scansione
	guardduty: Membri StartMonitoring
	guardduty: Membri StopMonitoring
	guardduty: UnarchiveFindings
	servizio di guardia: UpdateDetector
	servizio di guardia: UpdateFilter
	guardduty: Risposta UpdateFindings
	guardduty:UpdateIPSet
	servizio di guardia: UpdateMalware ScanSettings

Prefisso del servizio	Azioni
	guardduty: Rivelatori UpdateMember guardduty: Configurazione UpdateOrganization guardduty: Destinazione UpdatePublishing servizio di guardia: UpdateThreat IntelSet
healthlake	healthlake:CreateFHIRDatastore lago sanitario: CreateResource healthlake:DeleteFHIRDatastore lago sanitario: DeleteResource healthlake:DescribeFHIRDatastore Healthlake: descrivi FHIR ExportJob HealthLake: Descrivi Fhir ImportJob Healthlake: GetCapabilities healthlake:ListFHIRDatastores Lago sanitario: elenco FHIR ExportJobs Lago della salute: Elenco Fhir ImportJobs Lago della salute: ReadResource healthlake: Ottieni SearchWith healthlake: Pubblica SearchWith HealthLake: Start FHIR ExportJob Lago della salute: Startfhir ImportJob Lago della salute: UpdateResource

Prefisso del servizio	Azioni
honeycode	codice del miele: BatchCreate TableRows codice del miele: BatchDelete TableRows codice del miele: BatchUpdate TableRows codice del miele: BatchUpsert TableRows honeycode: Job DescribeTable DataImport honeycode: Dati GetScreen honeycode: automazione InvokeScreen honeycode: colonne ListTable honeycode: righe ListTable honeycode: ListTables honeycode: righe QueryTable honeycode: Job StartTable DataImport

Prefisso del servizio	Azioni
iam	iam: AddClient ID ID ToOpen ConnectProvider iam: AddRole ToInstance Profilo iam: AddUser ToGroup iam: AttachGroup Politica iam: AttachRole Politica iam: AttachUser Politica obiettivo: ChangePassword iam: CreateAccess Chiave iam: CreateAccount Alias Io sono: CreateGroup iam: CreateInstance Profilo iam: CreateLogin Profilo iam: CreateOpen ID ConnectProvider Io sono: CreatePolicy iam: CreatePolicy versione Sono: CreateRole iam:CreateSAMLProvider sono: CreateService LinkedRole Io sono: CreateService SpecificCredential sono: CreateUser nome: CreateVirtual mfaDevice

Prefisso del servizio	Azioni
	iam:DeactivateMFADevice iam: chiave DeleteAccess iam: DeleteAccount Alias Io sono: DeleteAccount PasswordPolicy iam: DeleteCloud FrontPublic Chiave obiettivo: DeleteGroup iam: DeleteGroup Politica iam: DeleteInstance Profilo iam: DeleteLogin Profilo iam: DeleteOpen ID ConnectProvider Io sono: DeletePolicy iam: DeletePolicy versione Sono: DeleteRole sono: DeleteRole PermissionsBoundary iam: DeleteRole Politica iam:DeleteSAMLProvider iam: DeleteServer Certificato Sono: DeleteService LinkedRole Io sono: DeleteService SpecificCredential iam: DeleteSigning Certificato IAM: elimina SSH PublicKey

Prefisso del servizio	Azioni
	Io sono: DeleteUser sono: DeleteUser PermissionsBoundary iam: DeleteUser Politica iam: DeleteVirtual MFADevice iam: Politica DetachGroup iam: DetachRole Politica iam: DetachUser Politica iam:EnableMFADevice iam: GenerateCredential rapporto Sono: GenerateOrganizations AccessReport iam: GenerateService LastAccessed Dettagli iam: GetAccess KeyLast Usato iam: GetAccount AuthorizationDetails Io sono: GetAccount EmailAddress iam: GetAccount Nome Sono: GetAccount PasswordPolicy iam: GetAccount Riepilogo iam: GetCloud FrontPublic Chiave obiettivo: GetContext KeysFor CustomPolicy Io sono: GetContext KeysFor PrincipalPolicy iam: GetCredential rapporto

Prefisso del servizio	Azioni
	Sono: GetGroup iam: GetGroup Politica iam: GetInstance Profilo iam: GetLogin Profilo iam:GetMFADevice iam: GetOpen ID ConnectProvider sono: GetOrganizations AccessReport sono: GetPolicy iam: GetPolicy versione Sono: GetRole iam: GetRole Politica iam:GetSAMLProvider iam: GetServer Certificato iam: GetService LastAccessed Dettagli iam: GetService LastAccessed DetailsWith Entità sono: GetService LinkedRole DeletionStatus Iam: getSSH PublicKey sono: GetUser iam: GetUser Politica iam: ListAccess Chiavi iam: ListAccount alias

Prefisso del servizio	Azioni
	sono: ListAttached GroupPolicies Io sono: ListAttached RolePolicies Io sono: ListAttached UserPolicies iam: ListCloud FrontPublic chiavi Io sono: ListEntities ForPolicy iam: ListGroup Politiche Io sono: ListGroups sono: ListGroups ForUser iam: ListInstance Profili iam: ListInstance ProfilesFor Ruolo iam:ListMFADevices iam: ListOpen ID ConnectProviders Io sono: ListPolicies iam: ListPolicies GrantingService Accesso iam: ListPolicy Versioni iam: ListRole Politiche Io sono: ListRoles iam:ListSAMLProviders iam: ListServer Certificati sono: ListService SpecificCredentials iam: ListSigning Certificati

Prefisso del servizio	Azioni
	IAM: elenca SSH PublicKeys IAM:Liststs Status RegionalEndpoints iam: Politiche ListUser Io sono: ListUsers iam: dispositivi ListVirtual MFA iam: Politica PutGroup obiettivo: PutRole PermissionsBoundary iam: PutRole Politica obiettivo: PutUser PermissionsBoundary iam: PutUser Politica iam: RemoveClient ID FromOpen ID ConnectProvider iam: RemoveRole FromInstance Profilo iam: RemoveUser FromGroup Io sono: ResetService SpecificCredential iam:ResyncMFADevice Io sono: SetDefault PolicyVersion iam: SetSecurity TokenService Preferenze iam:setSTS Status RegionalEndpoint iam: Politica SimulateCustom iam: SimulatePrincipal Politica iam: UpdateAccess Chiave

Prefisso del servizio	Azioni
	obiettivo: UpdateAccount EmailAddress iam: UpdateAccount Nome Sono: UpdateAccount PasswordPolicy Io sono: UpdateAssume RolePolicy iam: UpdateCloud FrontPublic Chiave obiettivo: UpdateGroup iam: UpdateLogin Profilo iam: UpdateOpen ConnectProvider Thumbprint ID Io sono: UpdateRole iam: UpdateRole Descrizione iam:UpdateSAMLProvider iam: UpdateServer Certificato Sono: UpdateService SpecificCredential iam: UpdateSigning Certificato IAM: aggiorna SSH PublicKey Io sono: UpdateUser iam: UploadCloud FrontPublic Chiave iam: UploadServer Certificato iam: UploadSigning Certificato IAM: carica SSH PublicKey

Prefisso del servizio	Azioni
identitystore	archivio di identità: CreateGroup identitystore: Iscrizione CreateGroup identitystore: CreateUser archivio di identità: DeleteGroup identitystore: Iscrizione DeleteGroup identitystore: DeleteUser archivio di identità: DescribeGroup identitystore: Iscrizione DescribeGroup identitystore: DescribeUser identitystore: ID GetGroup archivio di identità: GetGroup MembershipId identitystore: ID GetUser archivio di identità: IsMember InGroups identitystore: abbonamenti ListGroup identitystore: Membro ListGroup MembershipsFor identitystore: ListGroups archivio di identità: ListUsers archivio di identità: UpdateGroup archivio di identità: UpdateUser

Prefisso del servizio	Azioni
imagebuilder	imagebuilder: Creazione CancelImage imagebuilder: Esecuzione CancelLifecycle imagebuilder: CreateComponent imagebuilder: ricetta CreateContainer imagebuilder: configurazione CreateDistribution imagebuilder: CreateImage imagebuilder: Pipeline CreateImage imagebuilderCreateImage: Ricetta imagebuilder: configurazione CreateInfrastructure imagebuilder: politica CreateLifecycle imagebuilder: CreateWorkflow generatore di immagini: DeleteComponent imagebuilder: ricetta DeleteContainer imagebuilder: configurazione DeleteDistribution imagebuilder: DeleteImage imagebuilder: Pipeline DeleteImage imagebuilderDeleteImage: Ricetta imagebuilder: configurazione DeleteInfrastructure imagebuilder: politica DeleteLifecycle imagebuilder: DeleteWorkflow imagebuilder: politica GetComponent

Prefisso del servizio	Azioni
	imagebuilder: GetContainer RecipePolicy imagebuilder: politica GetImage imagebuilder: GetImage RecipePolicy imagebuilder: esecuzione GetLifecycle imagebuilder: politica GetLifecycle imagebuilder: Esecuzione GetWorkflow imagebuilder: GetWorkflow StepExecution generatore di immagini: ImportComponent imagebuilder: immagine ImportVm generatore di immagini: ListComponent BuildVersions generatore di immagini: ListComponents imagebuilder: ricette ListContainer imagebuilder: configurazioni ListDistribution imagebuilder: ListImage BuildVersions imagebuilder: pacchetti ListImage imagebuilder: ListImage PipelineImages imagebuilder: Pipeline ListImage imagebuilderListImage: ricette imagebuilder: ListImages imagebuilder: aggregazioni ListImage ScanFinding generatore di immagini: ListImage ScanFindings

Prefisso del servizio	Azioni
	imagebuilder: configurazioni ListInfrastructure imagebuilder: ListLifecycle ExecutionResources imagebuilder: Esecuzioni ListLifecycle imagebuilderListLifecycle: politiche imagebuilder: ListWaiting WorkflowSteps imagebuilder: Esecuzioni ListWorkflow generatore di immagini: ListWorkflows generatore di immagini: ListWorkflow StepExecutions imagebuilder: politica PutComponent imagebuilder: PutContainer RecipePolicy imagebuilder: politica PutImage imagebuilder: PutImage RecipePolicy generatore di immagini: SendWorkflow StepAction generatore di immagini: StartImage PipelineExecution generatore di immagini: StartResource StateUpdate imagebuilder: configurazione UpdateDistribution imagebuilder: Pipeline UpdateImage imagebuilder: UpdateInfrastructure Configurazione

Prefisso del servizio	Azioni
inspector	ispettore: AddAttributes ToFindings
	ispettore: Target CreateAssessment
	ispettore: modello CreateAssessment
	ispettore: Anteprima CreateExclusions
	ispettore: Gruppo CreateResource
	ispettore: Corri DeleteAssessment
	ispettore: Target DeleteAssessment
	ispettore: modello DeleteAssessment
	ispettore: corre DescribeAssessment
	ispettore: obiettivi DescribeAssessment
	ispettore: modelli DescribeAssessment
	ispettore: Ruolo DescribeCross AccountAccess
	ispettore: DescribeExclusions
	ispettore: DescribeFindings
	ispettore: gruppi DescribeResource
	inspector: Pacchetti DescribeRules
	ispettore: rapporto GetAssessment
	ispettore: Anteprima GetExclusions
	inspector: Metadati GetTelemetry
	ispettore: ListAssessment RunAgents
	ispettore: corre ListAssessment

Prefisso del servizio	Azioni
	ispettore: obiettivi ListAssessment
	ispettore: modelli ListAssessment
	inspector: Abbonamenti ListEvent
	ispettore: ListExclusions
	ispettore: ListFindings
	ispettore: Pacchetti ListRules
	ispettore: PreviewAgents
	ispettore: ruolo RegisterCross AccountAccess
	ispettore: RemoveAttributes FromFindings
	ispettore: corri StartAssessment
	ispettore: Esegui StopAssessment
	ispettore: Evento SubscribeTo
	ispettore: Evento UnsubscribeFrom
	ispettore: Target UpdateAssessment

Prefisso del servizio	Azioni
inspector2	ispettore 2: AssociateMember ispettore 2: BatchGet AccountStatus ispettore 2: BatchGet CodeSnippet ispettore 2: BatchGet FindingDetails inspector2: Informazioni BatchGet FreeTrial inspector2:2 Stato BatchGet MemberEc DeepInspection inspector2:2 Status BatchUpdate MemberEc DeepInspection isspector2: Rapporto CancelFindings ispettore 2: Esportazione CancelSbom ispettore 2: CreateCis ScanConfiguration ispettore 2: CreateFilter ispettore 2: rapporto CreateFindings ispettore 2: Esportazione CreateSbom ispettore 2: DeleteCis ScanConfiguration ispettore 2: DeleteFilter inspector2: Configurazione DescribeOrganization inspector2:Disable ispettore 2: DisableDelegated AdminAccount ispettore 2: DisassociateMember inspector2:Enable ispettore 2: EnableDelegated AdminAccount

Prefisso del servizio	Azioni
	ispettore 2: GetCis ScanReport inspector2: Dettagli GetCis ScanResult ispettore 2: GetConfiguration ispettore 2: GetDelegated AdminAccount inspector2:2 Configurazione GetEc DeepInspection inspector2: Chiave GetEncryption ispettore 2: GetFindings ReportStatus ispettore 2: GetMember ispettore 2: Esporta GetSbom inspector2: Autorizzazioni ListAccount ispettore 2: ListCis ScanConfigurations ispettore 2: controlli ListCis ScanResults AggregatedBy ispettore 2: ListCis ScanResults AggregatedBy TargetResource inspector2: scansioni ListCis ispettore 2: ListCoverage inspector2: Statistiche ListCoverage ispettore 2: ListDelegated AdminAccounts ispettore 2: ListFilters inspector2: Aggregazioni ListFinding ispettore 2: ListFindings ispettore 2: ListMembers

Prefisso del servizio	Azioni
	inspector2: totali ListUsage inspector2: ResetEncryption Chiave ispettore 2: SearchVulnerabilities ispettore 2: SendCis SessionHealth ispettore 2: SendCis SessionTelemetry ispettore 2: sessione StartCis inspector2: Sessione StopCis ispettore 2: UpdateCis ScanConfiguration ispettore 2: UpdateConfiguration inspector2:2 Configurazione UpdateEc DeepInspection inspector2: Chiave UpdateEncryption ispettore 2: UpdateFilter inspector2: Configurazione UpdateOrganization inspector2: Configurazione Ec2 UpdateOrg DeepInspection

Prefisso del servizio	Azioni
iot	iot: AcceptCertificate trasferimento
	iot: AddThing ToBilling Gruppo
	iot: AddThing ToThing Gruppo
	iot: AssociateTargets WithJob
	IoT: AttachPolicy
	iot: AttachPrincipal Politica
	iot: AttachSecurity Profilo
	iot: AttachThing Principale
	iot: CancelAudit MitigationActions Attività
	iot: CancelAudit Attività
	iot: CancelCertificate Trasferimento
	iot: CancelDetect MitigationActions Attività
	iot: CancelJob
	iot: CancelJob Esecuzione
	iot: ClearDefault Autorizzatore
	iot: ConfirmTopic RuleDestination
	iot: CreateAudit soppressione
	iot: CreateAuthorizer
	iot: CreateBilling Gruppo
	iot: CreateCertificate FromCsr
	iot: CreateCertificate Fornitore

Prefisso del servizio	Azioni
	iot: CreateCustom metrico
	iot: CreateDimension
	iot: CreateDomain configurazione
	iot: CreateDynamic ThingGroup
	iot: CreateFleet metrico
	iot: CreateJob
	iot: CreateJob Modello
	iot: CreateKeys AndCertificate
	iot: CreateMitigation Azione
	iot:CreateOTAUpdate
	iot: CreatePackage
	iot: CreatePackage versione
	iot: CreatePolicy
	iot: CreatePolicy versione
	iot: CreateProvisioning Reclamo
	iot: CreateProvisioning Modello
	iot: CreateProvisioning TemplateVersion
	iot: CreateRole alias
	iot: verifica CreateScheduled
	iot: CreateSecurity Profilo
	iot: CreateStream

Prefisso del servizio	Azioni
	IoT: CreateThing
	iot: CreateThing Gruppo
	iot: CreateThing Tipo
	iot: CreateTopic Regola
	iot: CreateTopic RuleDestination
	IoT: DeleteAccount AuditConfiguration
	iot: DeleteAudit soppressione
	iot: DeleteAuthorizer
	iot: DeleteBilling Gruppo
	iot:DeleteCACertificate
	iot: DeleteCertificate
	iot: DeleteCertificate Fornitore
	iot: DeleteCustom metrico
	iot: DeleteDimension
	iot: DeleteDomain configurazione
	iot: DeleteDynamic ThingGroup
	iot: DeleteFleet metrico
	iot: DeleteJob
	iot: DeleteJob Esecuzione
	iot: DeleteJob Modello
	iot: DeleteMitigation Azione

Prefisso del servizio	Azioni
	iot:DeleteOTAUpdate
	iot: DeletePackage
	iot: DeletePackage versione
	iot: DeletePolicy
	iot: DeletePolicy versione
	iot: DeleteProvisioning Modello
	iot: DeleteProvisioning TemplateVersion
	iot: DeleteRegistration Codice
	iot: DeleteRole Alias
	iot: verifica DeleteScheduled
	iot: DeleteSecurity Profilo
	iot: DeleteStream
	IoT: DeleteThing
	iot: DeleteThing Gruppo
	iot: DeleteThing Tipo
	iot: DeleteTopic Regola
	iot: DeleteTopic RuleDestination
	IoT: elimina V2 LoggingLevel
	iot: Tipo DeprecateThing
	iot: DescribeAccount AuditConfiguration
	iot: DescribeAudit ricerca

Prefisso del servizio	Azioni
	iot: DescribeAudit MitigationActions Attività
	iot: DescribeAudit soppressione
	iot: Attività DescribeAudit
	iot: DescribeAuthorizer
	iot: DescribeBilling Gruppo
	iot:DescribeCACertificate
	iot: DescribeCertificate
	iot: DescribeCertificate Fornitore
	iot: DescribeCustom metrico
	iot: Autorizzatore DescribeDefault
	iot: Attività DescribeDetect MitigationActions
	iot: DescribeDimension
	iot: DescribeDomain configurazione
	iot: DescribeEndpoint
	iot: DescribeEvent configurazioni
	iot: metrico DescribeFleet
	iot: DescribeIndex
	IoT: DescribeJob
	iot: DescribeJob Esecuzione
	iot: DescribeJob Modello
	iot: DescribeManaged JobTemplate

Prefisso del servizio	Azioni
	iot: DescribeMitigation Azione
	iot: DescribeProvisioning Modello
	iot: DescribeProvisioning TemplateVersion
	iot: DescribeRole alias
	iot: verifica DescribeScheduled
	iot: DescribeSecurity Profilo
	iot: DescribeStream
	IoT: DescribeThing
	iot: DescribeThing Gruppo
	iot: DescribeThing RegistrationTask
	iot: DescribeThing Tipo
	iot: DetachPolicy
	iot: DetachPrincipal Politica
	iot: DetachSecurity Profilo
	iot: DetachThing Principale
	iot: DisableTopic Regola
	iot: EnableTopic Regola
	iot: GetBehavior ModelTraining riassunti
	iot: Aggregazione GetBuckets
	iot: GetCardinality
	iot: GetEffective Politiche

Prefisso del servizio	Azioni
	iot: GetJob Documento
	iot: GetLogging Opzioni
	iot:GetOTAUpdate
	iot: GetPackage
	iot: GetPackage configurazione
	iot: GetPackage versione
	iot: GetPercentiles
	IoT: GetPolicy
	iot: GetPolicy versione
	iot: GetRegistration Codice
	iot: GetStatistics
	iot: GetTopic Regola
	iot: GetTopic RuleDestination
	IoT: getV2 LoggingOptions
	iot: violazioni ListActive
	iot: ListAttached Politiche
	iot: ListAudit risultati
	iot: ListAudit MitigationActions Esecuzioni
	iot: ListAudit MitigationActions Compiti
	iot: ListAudit Soppressioni
	iot: Compiti ListAudit

Prefisso del servizio	Azioni
	iot: ListAuthorizers
	iot: ListBilling Gruppi
	iot:ListCACertificates
	iot: ListCertificate Fornitori
	iot: ListCertificates
	iot: di ListCertificates CA
	iot: metriche ListCustom
	iot: ListDetect MitigationActions Esecuzioni
	iot: ListDetect MitigationActions Compiti
	iot: ListDimensions
	iot: ListDomain configurazioni
	iot: Metriche ListFleet
	iot: ListIndices
	iot: ListJob ExecutionsFor Job
	iot: ListJob ExecutionsFor Cosa
	iot: ListJobs
	iot: ListJob modelli
	iot: ListManaged JobTemplates
	iot: ListMetric Valori
	iot: ListMitigation Azioni
	iot:ListOTAUpdates

Prefisso del servizio	Azioni
	iot: ListOutgoing Certificati
	iot: ListPackages
	iot: ListPackage Versioni
	iot: ListPolicies
	iot: ListPolicy Principi
	iot: Versioni ListPolicy
	iot: ListPrincipal Politiche
	iot: ListPrincipal Cose
	iot: ListProvisioning Modelli
	iot: ListProvisioning TemplateVersions
	IoT: ListRelated ResourcesFor AuditFinding
	iot: ListRole alias
	iot: Audit ListScheduled
	iot: Profili ListSecurity
	iot: ListSecurity ProfilesFor Obiettivo
	iot: ListStreams
	IoT: ListTargets ForPolicy
	iot: ListTargets ForSecurity Profilo
	iot: ListThing Gruppi
	iot: ListThing GroupsFor Cosa
	iot: ListThing Principi

Prefisso del servizio	Azioni
	iot: Rapporti ListThing RegistrationTask
	iot: ListThing RegistrationTasks
	IoT: ListThings
	iot: ListThings InBilling Gruppo
	iot: ListThings InThing Gruppo
	iot: ListThing tipi
	iot: ListTopic RuleDestinations
	iot: ListTopic Regole
	IoT: listv2 LoggingLevels
	iot: Eventi ListViolation
	iot: PutVerification StateOn Violazione
	iot:RegisterCACertificate
	iot: RegisterCertificate
	iot: RegisterCertificate senza CA
	iot: RegisterThing
	iot: RejectCertificate Trasferimento
	iot: RemoveThing FromBilling Gruppo
	iot: RemoveThing FromThing Gruppo
	iot: ReplaceTopic Regola
	iot: SearchIndex
	iot: SetDefault Autorizzatore

Prefisso del servizio	Azioni
	iot: SetDefault PolicyVersion
	iot: SetLogging Opzioni
	IoT: setV2 LoggingLevel
	IoT: setV2 LoggingOptions
	iot: Attività StartAudit MitigationActions
	iot: StartDetect MitigationActions Attività
	iot: StartOn DemandAudit Attività
	iot: StartThing RegistrationTask
	IoT: StopThing RegistrationTask
	IoT: TestAuthorization
	iot: TestInvoke Autorizzatore
	iot: TransferCertificate
	IoT: UpdateAccount AuditConfiguration
	iot: UpdateAudit soppressione
	iot: UpdateAuthorizer
	iot: UpdateBilling Gruppo
	iot:UpdateCACertificate
	iot: UpdateCertificate
	iot: UpdateCertificate Fornitore
	iot: UpdateCustom metrico
	iot: UpdateDimension

Prefisso del servizio	Azioni
	iot: UpdateDomain configurazione
	iot: UpdateDynamic ThingGroup
	iot: UpdateEvent configurazioni
	iot: metrico UpdateFleet
	iot: configurazione UpdateIndexing
	iot: UpdateJob
	iot: UpdateMitigation Azione
	iot: UpdatePackage
	iot: UpdatePackage configurazione
	iot: UpdatePackage versione
	iot: UpdateProvisioning Modello
	iot: UpdateRole Alias
	iot: verifica UpdateScheduled
	iot: UpdateSecurity Profilo
	iot: UpdateStream
	IoT: UpdateThing
	iot: UpdateThing Gruppo
	iot: UpdateThing GroupsFor Cosa
	iot: UpdateTopic RuleDestination
	IoT: ValidateSecurity ProfileBehaviors

Prefisso del servizio	Azioni
iotanalytics	iotanalytics: rielaborazione CancelPipeline analisi IoT: CreateChannel analisi IoT: CreateDataset iotanalytics: Contenuto CreateDataset analisi IoT: CreateDatastore analisi IoT: CreatePipeline analisi IoT: DeleteChannel analisi IoT: DeleteDataset iotanalytics: Contenuto DeleteDataset analisi IoT: DeleteDatastore analisi IoT: DeletePipeline analisi IoT: DescribeChannel analisi IoT: DescribeDataset analisi IoT: DescribeDatastore iotanalytics: opzioni DescribeLogging analisi iot: DescribePipeline iotanalytics: Contenuto GetDataset analisi IoT: ListChannels iotanalytics: Contenuti ListDataset analisi IoT: ListDatasets analisi IoT: ListDatastores

Prefisso del servizio	Azioni
	analisi IoT: ListPipelines iotanalytics: opzioni PutLogging iotanalytics: Attività RunPipeline iotanalytics: dati SampleChannel iotanalytics: rielaborazione StartPipeline analisi IoT: UpdateChannel analisi IoT: UpdateDataset analisi IoT: UpdateDatastore analisi IoT: UpdatePipeline
iotdeviceadvisor	iotdeviceadvisor: definizione CreateSuite iotdeviceadvisor: DeleteSuite definizione iotdeviceadvisor: GetEndpoint iotdeviceadvisor: definizione GetSuite iotdeviceadvisor: GetSuite Esegui iotdeviceadvisor: GetSuite RunReport iotdeviceadvisor: definizioni ListSuite iotdeviceadvisor: ListSuite esegue iotdeviceadvisor: StartSuite Esegui iotdeviceadvisor: StopSuite Esegui iotdeviceadvisor: UpdateSuite definizione

Prefisso del servizio	Azioni
iotevents	iotevents: BatchAcknowledge allarme iotevents: rilevatore BatchDelete iotevents: Allarme BatchDisable iotevents: Allarme BatchEnable iotevents: Allarme BatchReset iotevents: Allarme BatchSnooze iotevents: rilevatore BatchUpdate iotevents: Modello CreateAlarm iotevents: Modello CreateDetector iotevents: CreateInput iotevents: Modello DeleteAlarm iotevents: Modello DeleteDetector iotevents: DeleteInput eventi IoT: DescribeAlarm iotevents: Modello DescribeAlarm iotevents: DescribeDetector iotevents: Modello DescribeDetector iotevents: DescribeDetector ModelAnalysis eventi IoT: DescribeInput iotevents: Opzioni DescribeLogging iotevents: Risultati GetDetector ModelAnalysis

Prefisso del servizio	Azioni
	iotevents: Modelli ListAlarm eventi iot: ListAlarm ModelVersions eventi IoT: ListAlarms iotevents: Modelli ListDetector eventi iot: ListDetector ModelVersions eventi IoT: ListDetectors iotevents: percorsi ListInput eventi iot: ListInputs iotevents: Opzioni PutLogging eventi iot: StartDetector ModelAnalysis iotevents: Modello UpdateAlarm iotevents: Modello UpdateDetector iotevents: UpdateInput
iotfleethub	hub iotfleet: CreateApplication hub iotfleet: DeleteApplication hub iotfleet: DescribeApplication hub iotfleet: ListApplications hub iotfleet: UpdateApplication

Prefisso del servizio	Azioni
iotsitewise	per quanto riguarda il sito IoT: AssociateAssets per quanto riguarda il sito IoT: AssociateTime SeriesTo AssetProp erty per quanto riguarda il sito IoT: BatchAssociate ProjectAssets per quanto riguarda il sito IoT: BatchDisassociate ProjectAssets iotsitewise: BatchGet AssetProperty Valore iotsitewise: BatchGet AssetProperty ValueHistory iotsitewise: BatchPut AssetProperty Valore iotsitewise: CreateAccess Politica iotsitewise: CreateAsset iotsitewise: Modello CreateAsset iotsitewise: CreateAsset ModelComposite Modello iotsitewise: CreateBulk ImportJob per quanto riguarda il sito IoT: CreateDashboard per quanto riguarda il sito IoT: CreateGateway per quanto riguarda il sito IoT: CreatePortal per quanto riguarda il sito IoT: CreateProject iotsitewise: Politica DeleteAccess iotsitewise: DeleteAsset iotsitewise: Modello DeleteAsset iotsitewise: DeleteAsset ModelComposite Modello

Prefisso del servizio	Azioni
	iotsitewise: DeleteDashboard per quanto riguarda il sito IoT: DeleteGateway per quanto riguarda il sito IoT: DeletePortal per quanto riguarda il sito IoT: DeleteProject iotsitewise: Serie DeleteTime iotsitewise: DescribeAccess Politica iotsitewise: DescribeAsset per quanto riguarda il sito IoT: DescribeAsset CompositeModel iotsitewise: Modello DescribeAsset iotsitewise: DescribeAsset ModelComposite Modello iotsitewise: DescribeAsset Proprietà iotsitewise: DescribeBulk ImportJob per quanto riguarda il sito IoT: DescribeDashboard per quanto riguarda il sito IoT: DescribeDefault EncryptionConfiguration per quanto riguarda il sito IoT: DescribeGateway per quanto riguarda il sito IoT: DescribeGateway CapabilityConfiguration iotsitewise: Opzioni DescribeLogging iotsitewise: DescribePortal per quanto riguarda il sito IoT: DescribeProject iotsitewise: Configurazione DescribeStorage

Prefisso del servizio	Azioni
	iotsitewise: DescribeTime Serie iotsitewise: DisassociateAssets per quanto riguarda il sito IoT: DisassociateTime SeriesFrom AssetProperty per quanto riguarda il sito IoT: ExecuteAction per quanto riguarda il sito IoT: ExecuteQuery iotsitewise: Politiche ListAccess iotsitewise: ListActions iotsitewise: Modelli ListAsset ModelComposite iotsitewise: ListAsset ModelProperties iotsitewise: Modelli ListAsset iotsitewise: ListAsset Proprietà iotsitewise: ListAsset Relazioni iotsitewise: ListAssets iotsitewise: Risorse ListAssociated iotsitewise: ListBulk ImportJobs iotsitewise: ListComposition Relazioni iotsitewise: ListDashboards per quanto riguarda il sito IoT: ListGateways per quanto riguarda il sito IoT: ListPortals iotsitewise: Risorse ListProject

Prefisso del servizio	Azioni
	iotsitewise: ListProjects iotsitewise: Serie ListTime iotsitewise: PutDefault EncryptionConfiguration iotsitewise: Opzioni PutLogging iotsitewise: PutStorage Configurazione iotsitewise: UpdateAccess Politica iotsitewise: UpdateAsset iotsitewise: Modello UpdateAsset iotsitewise: UpdateAsset ModelComposite Modello iotsitewise: UpdateAsset Proprietà iotsitewise: UpdateDashboard per quanto riguarda il sito IoT: UpdateGateway per quanto riguarda il sito IoT: UpdateGateway CapabilityConfigur ation per quanto riguarda il sito IoT: UpdatePortal per quanto riguarda il sito IoT: UpdateProject

Prefisso del servizio	Azioni
iottwinmaker	iottwinmaker: CancelMetadata TransferJob
	iottwinmaker: Tipo CreateComponent
	iottwinmaker: CreateEntity
	iottwinmaker: CreateMetadata TransferJob
	iottwinmaker: CreateScene
	iottwinmaker: Job CreateSync
	iottwinmaker: CreateWorkspace
	iottwinmaker: Tipo DeleteComponent
	iottwinmaker: DeleteEntity
	iottwinmaker: DeleteScene
	iottwinmaker: Job DeleteSync
	iottwinmaker: DeleteWorkspace
	iottwinmaker: ExecuteQuery
	iottwinmaker: GetMetadata TransferJob
	iottwinmaker: Piano GetPricing
	iottwinmaker: GetScene
	iottwinmaker: Job GetSync
	iottwinmaker: ListComponents
	iottwinmaker: tipi ListComponent
	iottwinmaker: ListEntities
	iottwinmaker: ListMetadata TransferJobs

Prefisso del servizio	Azioni
	iottwinmaker: ListProperties
	iottwinmaker: ListScenes
	iottwinmaker: ListSync Offerte di lavoro
	iottwinmaker: ListSync Risorse
	iottwinmaker: ListWorkspaces
	iottwinmaker: Tipo UpdateComponent
	iottwinmaker: UpdateEntity
	iottwinmaker: Piano UpdatePricing
	iottwinmaker: UpdateScene
	iottwinmaker: UpdateWorkspace

Prefisso del servizio	Azioni
iotwireless	IoT senza fili: AssociateAws AccountWith PartnerAccount IoT wireless: AssociateMulticast GroupWith FuotaTask IoT wireless: AssociateWireless DeviceWith FuotaTask IoT wireless: AssociateWireless DeviceWith MulticastGroup iotwireless: Cosa AssociateWireless DeviceWith iotwireless: certificato AssociateWireless GatewayWith iotwireless: Cosa AssociateWireless GatewayWith iotwireless: CancelMulticast GroupSession IoT wireless: CreateDestination iotwireless: Profilo CreateDevice iotwireless: Attività CreateFuota iotwireless: Gruppo CreateMulticast iotwireless: CreateNetwork AnalyzerConfiguration iotwireless: Profilo CreateService iotwireless: Dispositivo CreateWireless iotwireless: gateway CreateWireless IoT wireless: CreateWireless GatewayTask iotwireless: definizione CreateWireless GatewayTask iotwireless: DeleteDestination iotwireless: Profilo DeleteDevice iotwireless: Attività DeleteFuota

Prefisso del servizio	Azioni
	iotwireless: Gruppo DeleteMulticast iotwireless: DeleteNetwork AnalyzerConfiguration iotwireless: messaggi DeleteQueued iotwireless: Profilo DeleteService iotwireless: Dispositivo DeleteWireless iotwireless: Attività DeleteWireless DeviceImport iotwireless: gateway DeleteWireless IoT wireless: DeleteWireless GatewayTask iotwireless: definizione DeleteWireless GatewayTask iotwireless: dispositivo DeregisterWireless iotwireless: DisassociateAws AccountFrom PartnerAccount IoT wireless: DisassociateMulticast GroupFrom FuotaTask IoT wireless: DisassociateWireless DeviceFrom FuotaTask IoT wireless: DisassociateWireless DeviceFrom MulticastGroup iotwireless: Cosa DisassociateWireless DeviceFrom iotwireless: certificato DisassociateWireless GatewayFrom iotwireless: Cosa DisassociateWireless GatewayFrom iotwireless: GetDestination iotwireless: Profilo GetDevice iotwireless: GetEvent ConfigurationBy ResourceTypes iotwireless: Attività GetFuota

Prefisso del servizio	Azioni
	iotwireless: GetLog LevelsBy ResourceTypes iotwireless: configurazione GetMetric iotwireless: GetMetrics iotwireless: Gruppo GetMulticast iotwireless: GetMulticast GroupSession IoT wireless: GetNetwork AnalyzerConfiguration iotwireless: Conto GetPartner iotwireless: GetPosition iotwireless: configurazione GetPosition iotwireless: stima GetPosition iotwireless: GetResource EventConfiguration IoT wireless: GetResource LogLevel iotwireless: posizione GetResource iotwireless: punto finale GetService iotwireless: Profilo GetService iotwireless: Dispositivo GetWireless iotwireless: Attività GetWireless DeviceImport iotwireless: GetWireless DeviceStatistics iotwireless: gateway GetWireless IoT wireless: GetWireless GatewayCertificate iotwireless: Informazioni GetWireless GatewayFirmware

Prefisso del servizio	Azioni
	iotwireless: GetWireless GatewayStatistics IoT wireless: GetWireless GatewayTask iotwireless: definizione GetWireless GatewayTask iotwireless: ListDestinations iotwireless: Profili ListDevice iotwireless: Attività ListDevices ForWireless DeviceImport iotwireless: Configurazioni ListEvent iotwireless: Attività ListFuota iotwireless: Gruppi ListMulticast iotwireless: ListMulticast GroupsBy FuotaTask IoT wireless: ListNetwork AnalyzerConfigurations iotwireless: conti ListPartner iotwireless: configurazioni ListPosition iotwireless: Messaggi ListQueued iotwireless: Profili ListService iotwireless: Attività ListWireless DeviceImport iotwireless: Dispositivi ListWireless iotwireless: gateway ListWireless iotwireless: definizioni ListWireless GatewayTask iotwireless: Configurazione PutPosition iotwireless: PutResource LogLevel

Prefisso del servizio	Azioni
	iotwireless: livelli ResetAll ResourceLog iotwireless: ResetResource LogLevel iotwireless: Gruppo SendData ToMulticast iotwireless: dispositivo SendData ToWireless iotwireless: StartBulk AssociateWireless DeviceWith MulticastGroup IoT wireless: StartBulk DisassociateWireless DeviceFrom Multicast Group iotwireless: Attività StartFuota iotwireless: StartMulticast GroupSession IoT wireless: StartNetwork AnalyzerStream IoT wireless: StartSingle WirelessDevice ImportTask iotwireless: Attività StartWireless DeviceImport iotwireless: dispositivo TestWireless iotwireless: UpdateDestination IoT wireless: UpdateEvent ConfigurationBy ResourceTypes iotwireless: Attività UpdateFuota iotwireless: UpdateLog LevelsBy ResourceTypes iotwireless: configurazione UpdateMetric iotwireless: Gruppo UpdateMulticast iotwireless: UpdateNetwork AnalyzerConfiguration iotwireless: Conto UpdatePartner

Prefisso del servizio	Azioni
	iotwireless: UpdatePosition IoT wireless: UpdateResource EventConfiguration iotwireless: posizione UpdateResource iotwireless: dispositivo UpdateWireless iotwireless: Attività UpdateWireless DeviceImport iotwireless: gateway UpdateWireless

Prefisso del servizio	Azioni
ivs	ivs: Canale BatchGet ivs: BatchGet StreamKey ivs: revoca BatchStart ViewerSession ivs: CreateChannel ivs: Configurazione CreateEncoder ivs: Token CreateParticipant è: CreatePlayback RestrictionPolicy ivs: Configurazione CreateRecording ivs: Configurazione CreateStorage ivs: Chiave CreateStream ivs: DeleteChannel ivs: Configurazione DeleteEncoder ivs: DeletePlayback KeyPair è: DeletePlayback RestrictionPolicy ivs: Configurazione DeleteRecording ivs: Configurazione DeleteStorage ivs: Chiave DeleteStream ivs: DisconnectParticipant è: GetChannel è: GetComposition ivs: Configurazione GetEncoder

Prefisso del servizio	Azioni
	ivs: GetParticipant è: GetPlayback KeyPair è: GetPlayback RestrictionPolicy ivs: Configurazione GetRecording ivs: Configurazione GetStorage ivs: GetStream ivs: Chiave GetStream ivs: Sessione GetStream ivs: ImportPlayback KeyPair è: ListChannels è: ListCompositions ivs: Configurazioni ListEncoder ivs: Eventi ListParticipant ivs: ListParticipants è: ListPlayback KeyPairs è: ListPlayback RestrictionPolicies ivs: Configurazioni ListRecording ivs: Configurazioni ListStorage ivs: Chiavi ListStream ivs: ListStreams ivs: Sessioni ListStream

Prefisso del servizio	Azioni
	ivs: PutMetadata è: StartComposition è: StartViewer SessionRevocation è: StopComposition è: StopStream è: UpdateChannel è: UpdatePlayback RestrictionPolicy
ivschat	ivschat: Token CreateChat ivschat: Configurazione CreateLogging ivschat: CreateRoom ivschat: Configurazione DeleteLogging ivschat: DeleteMessage vista chat: DeleteRoom vista chat: DisconnectUser ivschat: Configurazione GetLogging ivschat: GetRoom ivschat: configurazioni ListLogging ivschat: ListRooms vista chat: SendEvent ivschat: Configurazione UpdateLogging ivschat: UpdateRoom

Prefisso del servizio	Azioni
kafka	kafka: BatchAssociate ScramSecret caffè: BatchDisassociate ScramSecret caffè: CreateCluster kafka: V2 CreateCluster kafka: CreateConfiguration caffè: CreateReplicator kafka: connessione CreateVpc kafka: DeleteCluster kafka: politica DeleteCluster kafka: DeleteConfiguration caffè: DeleteReplicator kafka: connessione DeleteVpc kafka: DescribeCluster kafka: operazione DescribeCluster kafka: Operazione V2 DescribeCluster kafkaDescribeCluster: V2 kafka: DescribeConfiguration kafka: revisione DescribeConfiguration kafka: Connessione DescribeVpc kafka: Broker GetBootstrap kafka: Politica GetCluster

Prefisso del servizio	Azioni
	kafka: GetCompatible KafkaVersions
	caffè: ListClient VpcConnections
	kafka: Operazioni ListCluster
	kafka: OperationsV2 ListCluster
	kafka: ListClusters
	kafka: V2 ListClusters
	kafka: revisioni ListConfiguration
	kafka: ListConfigurations
	kafka: versioni ListKafka
	kafka: ListNodes
	caffè: ListReplicators
	kafka: segreti ListScram
	kafka: Connessioni ListVpc
	kafka: Politica PutCluster
	kafka: RebootBroker
	caffè: RejectClient VpcConnection
	kafka: Conta UpdateBroker
	kafka: conservazione UpdateBroker
	kafka: tipo UpdateBroker
	kafka: Configurazione UpdateCluster
	kafka: UpdateCluster KafkaVersion

Prefisso del servizio	Azioni
	caffè: UpdateConfiguration caffè: UpdateConnectivity caffè: UpdateMonitoring kafka: Informazioni UpdateReplication kafka: UpdateSecurity caffè: UpdateStorage
kafkaconnect	connessione kafka: CreateConnector kafkaconnect: Plugin CreateCustom kafkaconnect: Configurazione CreateWorker kafkaconnect: DeleteConnector kafkaconnect: Plugin DeleteCustom kafkaconnect: Configurazione DeleteWorker kafkaconnect: DescribeConnector kafkaconnect: Plugin DescribeCustom kafkaconnect: Configurazione DescribeWorker kafkaconnect: ListConnectors kafkaconnect: Plugin ListCustom kafkaconnect: Configurazioni ListWorker kafkaconnect: UpdateConnector

Prefisso del servizio	Azioni
kendra	kendra: AssociateEntities ToExperience
	kendra: AssociatePersonas ToEntities
	kendra: Documento BatchDelete
	kendra: Serie BatchDelete FeaturedResults
	kendra: BatchGet DocumentStatus
	kendra: Documento BatchPut
	kendra: Suggerimenti ClearQuery
	kendra: CreateAccess ControlConfiguration
	kendra: Fonte CreateData
	kendra: CreateExperience
	kendra: CreateFaq
	kendra: CreateFeatured ResultsSet
	kendra: CreateIndex
	kendra: Elenco CreateQuery SuggestionsBlock
	kendra: CreateThesaurus
	kendra: Fonte DeleteData
	kendra: DeleteExperience
	kendra: DeleteFaq
	kendra: DeleteIndex
	kendra: mappatura DeletePrincipal
	kendra: Elenco DeleteQuery SuggestionsBlock

Prefisso del servizio	Azioni
	kendra: DeleteThesaurus
	kendra: DescribeAccess ControlConfiguration
	kendra: Fonte DescribeData
	kendra: DescribeExperience
	kendra: DescribeFaq
	kendra: DescribeFeatured ResultsSet
	kendra: DescribeIndex
	kendra: mappatura DescribePrincipal
	kendra: Elenco DescribeQuery SuggestionsBlock
	kendra: DescribeQuery SuggestionsConfig
	kendra: DescribeThesaurus
	kendra: DisassociateEntities FromExperience
	kendra: DisassociatePersonas FromEntities
	kendra: Suggerimenti GetQuery
	kendra: GetSnapshots
	kendra: ListAccess ControlConfigurations
	kendra: Fonti ListData
	kendra: Offerte di lavoro ListData SourceSync
	kendra: Persone ListEntity
	kendra: Entità ListExperience
	kendra: ListExperiences

Prefisso del servizio	Azioni
	kendra: ListFaqs
	kendra: ListFeatured ResultsSets
	kendra: ListGroups OlderThan OrderingId
	kendra: ListIndices
	kendra: Liste ListQuery SuggestionsBlock
	kendra: ListThesauri
	kendra: mappatura PutPrincipal
	kendra:Query
	kendra:Retrieve
	kendra: Job StartData SourceSync
	kendra: Job StopData SourceSync
	kendra: SubmitFeedback
	kendra: Fonte UpdateData
	kendra: UpdateExperience
	kendra: UpdateFeatured ResultsSet
	kendra: UpdateIndex
	kendra: Elenco UpdateQuery SuggestionsBlock
	kendra: UpdateQuery SuggestionsConfig
	kendra: UpdateThesaurus

Prefisso del servizio	Azioni
kinesis	cinesi: CreateStream cinesi: DecreaseStream RetentionPeriod cinesi: DeleteStream kinesis: Consumatore DeregisterStream cinesi: DescribeLimits cinesi: DescribeStream kinesis: Consumatore DescribeStream kinesis: Riepilogo DescribeStream kinesis: Monitoraggio DisableEnhanced kinesis: Monitoraggio EnableEnhanced cinesi: IncreaseStream RetentionPeriod cinesi: ListShards kinesis: Consumatori ListStream kinesis: ListStreams cinesi: MergeShards kinesis: Consumatore RegisterStream cinesi: SplitShard kinesis: crittografia StartStream kinesis: crittografia StopStream kinesis: Conta UpdateShard kinesis: Modalità UpdateStream

Prefisso del servizio	Azioni
kinesisanalytics	kinesis analytics: AddApplication CloudWatch LoggingOption
	kinesisanalytics: Input AddApplication
	kinesisanalytics: Configurazione AddApplication InputProcessing
	kinesisanalytics: Uscita AddApplication
	kinesisanalytics: Fonte AddApplication ReferenceData
	kinesisanalytics: AddApplication VpcConfiguration
	kinesisanalytics: CreateApplication
	kinesisanalytics: CreateApplication PresignedUrl
	kinesisanalytics: istantanea CreateApplication
	kinesisanalytics: DeleteApplication
	kinesisanalytics: DeleteApplication CloudWatch LoggingOption
	kinesisanalytics: Configurazione DeleteApplication InputProcessing
	kinesisanalytics: Uscita DeleteApplication
	kinesisanalytics: Fonte DeleteApplication ReferenceData
	kinesisanalytics: Istantanea DeleteApplication
	kinesisanalytics: DeleteApplication VpcConfiguration
	kinesisanalytics: DescribeApplication
	kinesisanalytics: istantanea DescribeApplication
	kinesisanalyticsDescribeApplication: Versione
	kinesisanalytics: Schema DiscoverInput
	kinesisanalytics: ListApplications

Prefisso del servizio	Azioni
	kinesisanalytics: istantanee ListApplication kinesisanalytics: ListApplication Versioni kinesisanalytics: RollbackApplication kinesisanalytics: StartApplication kinesisanalytics: StopApplication kinesisanalytics: UpdateApplication kinesisanalytics: UpdateApplication MaintenanceConfiguration

Prefisso del servizio	Azioni
kms	kms: cancellazione CancelKey kms: ConnectCustom KeyStore km: CreateAlias km: CreateCustom KeyStore km: CreateGrant km: CreateKey kms:Decrypt km: DeleteAlias km: DeleteCustom KeyStore km: DeleteImported KeyMaterial km: DescribeCustom KeyStores km: DescribeKey km: DisableKey kms: rotazione DisableKey km: DisconnectCustom KeyStore km: EnableKey kms: rotazione EnableKey kms:Encrypt kms: chiave GenerateData km: GenerateData KeyPair km: GenerateData KeyPair WithoutPlaintext

Prefisso del servizio	Azioni
	kms: testo semplice GenerateData KeyWithout km: GenerateMac km: GenerateRandom kms: Politica GetKey km: GetKey RotationStatus km: GetParameters ForImport kms: chiave GetPublic kms: Materiale ImportKey km: ListAliases km: ListGrants kms: politiche ListKey km: ListKeys km: sovvenzioni ListRetirable km: ReplicateKey km: RetireGrant km: RevokeGrant kms: cancellazione ScheduleKey kms:Sign kms: UpdateAlias km: UpdateCustom KeyStore kms: Descrizione UpdateKey

Prefisso del servizio	Azioni
	kms: Regione UpdatePrimary kms:Verify km: VerifyMac

Prefisso del servizio	Azioni
lambda	lambda: AddLayer VersionPermission
	lambda: AddLayer VersionPermission
	lambda: AddPermission
	lambda: AddPermission
	lambda: AddPermission
	lambda: CreateAlias
	lambda: CreateAlias
	lambda: CreateCode SigningConfig
	lambda: CreateEvent SourceMapping
	lambda: CreateEvent SourceMapping
	lambda: CreateFunction
	lambda: CreateFunction
	lambda: CreateFunction UrlConfig
	lambda: DeleteAlias
	lambda: DeleteAlias
	lambda: DeleteCode SigningConfig
	lambda: DeleteEvent SourceMapping
	lambda: DeleteEvent SourceMapping
	lambda: DeleteFunction
	lambda: DeleteFunction
	lambda: Config DeleteFunction CodeSigning

Prefisso del servizio	Azioni
	lambda: Concorrenza DeleteFunction
	lambda: Concorrenza DeleteFunction
	lambda: Config DeleteFunction EventInvoke
	lambda: DeleteFunction UrlConfig
	lambda: versione DeleteLayer
	lambda: versione DeleteLayer
	lambda: DeleteProvisioned ConcurrencyConfig
	lambda: impostazioni GetAccount
	lambda: impostazioni GetAccount
	lambda: GetAlias
	lambda: GetAlias
	lambda: GetCode SigningConfig
	lambda: GetEvent SourceMapping
	lambda: GetEvent SourceMapping
	lambda: GetFunction
	lambda: GetFunction
	lambda: GetFunction
	lambda: Config GetFunction CodeSigning
	lambda: Concorrenza GetFunction
	lambda: configurazione GetFunction
	lambda: configurazione GetFunction

Prefisso del servizio	Azioni
	lambda: configurazione GetFunction
	lambda: Config GetFunction EventInvoke
	lambda: GetFunction UrlConfig
	lambda: versione GetLayer
	lambda: versione GetLayer
	lambda: versione GetLayer
	lambda: versione GetLayer
	lambda: GetLayer VersionPolicy
	lambda: GetLayer VersionPolicy
	lambda: GetPolicy
	lambda: GetPolicy
	lambda: GetPolicy
	lambda: GetProvisioned ConcurrencyConfig
	lambda: GetRuntime ManagementConfig
	lambda: ListAliases
	lambda: ListAliases
	lambda: ListCode SigningConfigs
	lambda: ListEvent SourceMappings
	lambda: ListEvent SourceMappings
	lambda: configurazioni ListFunction EventInvoke
	lambda: ListFunctions

Prefisso del servizio	Azioni
	lambda: ListFunctions
	lambda: ListFunctions ByCode SigningConfig
	lambda: ListFunction UrlConfigs
	lambda: ListLayers
	lambda: ListLayers
	lambda: versioni ListLayer
	lambda: versioni ListLayer
	lambda: ListProvisioned ConcurrencyConfigs
	lambda: ListVersions ByFunction
	lambda: ListVersions ByFunction
	lambda: versione PublishLayer
	lambda: versione PublishLayer
	lambda: PublishVersion
	lambda: PublishVersion
	lambda: Config PutFunction CodeSigning
	lambda: Concorrenza PutFunction
	lambda: Concorrenza PutFunction
	lambda: Config PutFunction EventInvoke
	lambda: PutProvisioned ConcurrencyConfig
	lambda: PutRuntime ManagementConfig
	lambda: RemoveLayer VersionPermission

Prefisso del servizio	Azioni
	lambda: RemoveLayer VersionPermission
	lambda: RemovePermission
	lambda: RemovePermission
	lambda: RemovePermission
	lambda: UpdateAlias
	lambda: UpdateAlias
	lambda: UpdateCode SigningConfig
	lambda: UpdateEvent SourceMapping
	lambda: UpdateEvent SourceMapping
	lambda: Codice UpdateFunction
	lambda: Codice UpdateFunction
	lambda: Codice UpdateFunction
	lambda: configurazione UpdateFunction
	lambda: configurazione UpdateFunction
	lambda: configurazione UpdateFunction
	lambda: Config UpdateFunction EventInvoke
	lambda: UpdateFunction UrlConfig

Prefisso del servizio	Azioni
lex	lex: Oggetto BatchCreate CustomVocabulary lex: BatchDelete CustomVocabulary Oggetto lex: BatchUpdate CustomVocabulary Oggetto lex: BuildBot Locale lex: CreateBot alias lex: versione CreateBot lex: CreateExport lex: CreateIntent versione lex: CreateResource Politica Flex: CreateSlot TypeVersion lex: CreateTest SetDiscrepancy rapporto lex: CreateUpload Url Flex: DeleteBot lex: DeleteBot ChannelAssociation lex: DeleteExport lex: DeleteImport lex: DeleteIntent versione lex: DeleteResource Politica Flex: DeleteSlot TypeVersion lex: DeleteTest Set Flex: DeleteUtterances

Prefisso del servizio	Azioni
	lex: DescribeBot alias lex: raccomandazione DescribeBot Flex: DescribeBot ResourceGeneration lex: DescribeBot versione lex: DescribeCustom VocabularyMetadata lex: DescribeExport lex: DescribeImport lex: DescribeResource Politica lex: DescribeTest Esecuzione lex: DescribeTest Imposta lex: DescribeTest SetDiscrepancy rapporto lex: DescribeTest SetGeneration lex: GenerateBot elemento Flex: GetBot lex: GetBot alias lex: alias GetBot lex: GetBot ChannelAssociation lex: GetBot ChannelAssociations lex: GetBots lex: GetBot Versioni lex: GetBuiltin Intento

Prefisso del servizio	Azioni
	lex: GetBuiltin Intenti lex: GetBuiltin SlotTypes lex: GetExport lex: GetImport lex: GetIntent lex: GetIntents lex: GetIntent Versioni lex: GetMigration lex: GetMigrations lex: GetSlot tipo lex: GetSlot tipi lex: GetSlot TypeVersions lex: GetTest ExecutionArtifacts URL lex: GetUtterances Visualizza lex: ListBot alias lex: ListBot Raccomandazioni lex: ListBot ResourceGenerations lex: ListBots lex: ListBot Versioni lex: ListBuilt InIntents lex: ListBuilt InSlot tipi

Prefisso del servizio	Azioni
	lex: ListCustom VocabularyItems lex: ListExports lex: ListImports lex: ListIntent metriche lex: Percorsi ListIntent lex: ListRecommended Intenti lex: ListSession AnalyticsData lex: ListSession metriche lex: articoli ListTest ExecutionResult lex: ListTest Esecuzioni lex: set ListTest lex: PutBot lex: PutBot alias Flex: PutIntent lex: PutSlot tipo lex: SearchAssociated trascrizioni lex: raccomandazione StartBot Flex: StartImport lex: StartMigration lex: StartTest esecuzione lex: StartTest SetGeneration

Prefisso del servizio	Azioni
	lex: StopBot raccomandazione lex: UpdateBot alias lex: raccomandazione UpdateBot Flex: UpdateExport lex: UpdateResource Politica
license-manager-linux-subscriptions	license-manager-linux-subscriptions: Impostazioni GetService abbonamenti license-manager-linux-: ListLinux SubscriptionInstances license-manager-linux-subscriptions: Abbonamenti ListLinux license-manager-linux-subscriptions: Impostazioni UpdateService

Prefisso del servizio	Azioni
lightsail	lightsail: Ip AllocateStatic vela leggera: AttachCertificate ToDistribution vela leggera: AttachDisk lightsail: Bilanciatore AttachInstances ToLoad lightsail: Certificato AttachLoad BalancerTls lightsail: Ip AttachStatic vela leggera: CloseInstance PublicPorts vela leggera: CopySnapshot vela leggera: CreateBucket vela leggera: CreateBucket AccessKey vela leggera: CreateCertificate vela leggera: CreateCloud FormationStack lightsail: metodo CreateContact lightsail: Assistenza CreateContainer lightsail: CreateContainer ServiceDeployment lightsail: Accedi CreateContainer ServiceRegistry lightsail: CreateDisk vela leggera: CreateDisk FromSnapshot lightsail: istantanea CreateDisk lightsail: CreateDistribution vela leggera: CreateDomain

Prefisso del servizio	Azioni
	LightSail: crea dettagli GUI SessionAccess vela leggera: CreateInstances vela leggera: CreateInstances FromSnapshot lightsail: istantanea CreateInstance lightsail: coppia CreateKey lightsail: Balancer CreateLoad lightsail: Certificato CreateLoad BalancerTls lightsail: Banca dati CreateRelational lightsail: istantanea CreateRelational DatabaseFrom lightsail: CreateRelational DatabaseSnapshot vela leggera: DeleteAlarm lightsail: istantanea DeleteAuto lightsail: DeleteBucket vela leggera: DeleteBucket AccessKey vela leggera: DeleteCertificate lightsail: metodo DeleteContact lightsail: Immagine DeleteContainer lightsail: Assistenza DeleteContainer lightsail: DeleteDisk lightsail: istantanea DeleteDisk lightsail: DeleteDistribution

Prefisso del servizio	Azioni
	vela leggera: DeleteDomain lightsail: ingresso DeleteDomain lightsail: DeleteInstance lightsail: istantanea DeleteInstance lightsail: coppia DeleteKey lightsail: DeleteKnown HostKeys lightsail: Bilanciatore DeleteLoad lightsail: Certificato DeleteLoad BalancerTls lightsail: Banca dati DeleteRelational lightsail: DeleteRelational DatabaseSnapshot vela leggera: DetachCertificate FromDistribution vela leggera: DetachDisk lightsail: Bilanciatore DetachInstances FromLoad lightsail: Ip DetachStatic lightsail: Attivato DisableAdd lightsail: DownloadDefault KeyPair lightsail: Attivato EnableAdd lightsail: ExportSnapshot lightsail: nomi GetActive lightsail: GetAlarms lightsail: istantanee GetAuto

Prefisso del servizio	Azioni
	lightsail: GetBlueprints vela leggera: GetBucket AccessKeys lightsail: pacchetti GetBucket lightsail: GetBucket MetricData vela leggera: GetBuckets vela leggera: GetBundles vela leggera: GetCertificates lightsail: Record GetCloud FormationStack lightsail: metodi GetContact lightsail: dati API GetContainer lightsail: GetContainer immagini lightsail: Registra GetContainer lightsail: GetContainer ServiceDeployments lightsail: dati GetContainer ServiceMetric lightsail: GetContainer ServicePowers lightsail: Servizi GetContainer lightsail: Stima GetCost lightsail: GetDisk vela leggera: GetDisks lightsail: istantanea GetDisk lightsail: istantanee GetDisk

Prefisso del servizio	Azioni
	lightsail: Pacchetti GetDistribution lightsail: Ripristina GetDistribution LatestCache lightsail: GetDistribution MetricData vela leggera: GetDistributions vela leggera: GetDomain vela leggera: GetExport SnapshotRecords vela leggera: GetInstance vela leggera: GetInstance MetricData vela leggera: GetInstance PortStates vela leggera: GetInstances lightsail: istantanea GetInstance lightsail: istantanee GetInstance lightsail: Stato GetInstance lightsail: coppia GetKey lightsail: coppie GetKey lightsail: Balancer GetLoad lightsail: Dati GetLoad BalancerMetric lightsail: Bilanciatori GetLoad lightsail GetLoadBalancerTls: Certificati lightsail: Politiche GetLoad BalancerTls lightsail: GetOperation

Prefisso del servizio	Azioni
	vela leggera: GetOperations vela leggera: GetOperations ForResource vela leggera: GetRegions lightsail: Banca dati GetRelational lightsail: GetRelational DatabaseBlueprints vela leggera: GetRelational DatabaseBundles vela leggera: GetRelational DatabaseEvents lightsail: Eventi GetRelational DatabaseLog lightsail: Streams GetRelational DatabaseLog lightsail: GetRelational DatabaseMaster UserPassword lightsail: dati GetRelational DatabaseMetric lightsail: GetRelational DatabaseParameters lightsail: Database GetRelational lightsail: GetRelational DatabaseSnapshot vela leggera: GetRelational DatabaseSnapshots lightsail: Storia GetSetup lightsail: Ip GetStatic vela leggera: Ips GetStatic lightsail: coppia ImportKey lightsail: Peered IsVpc vela leggera: OpenInstance PublicPorts

Prefisso del servizio	Azioni
	vela leggera: PeerVpc vela leggera: PutAlarm vela leggera: PutInstance PublicPorts vela leggera: RebootInstance lightsail: Banca dati RebootRelational lightsail: Immagine RegisterContainer lightsail: Ip ReleaseStatic lightsail: cache ResetDistribution lightsail: SendContact MethodVerification vela leggera: SetIp AddressType lightsail: Secchio SetResource AccessFor lightsail: Https SetupInstance lightsail:StartGUISession vela leggera: StartInstance lightsail: Banca dati StartRelational lightsail:StopGUISession lightsail: StopInstance lightsail: Banca dati StopRelational lightsail: TestAlarm vela leggera: UnpeerVpc vela leggera: UpdateBucket

Prefisso del servizio	Azioni
	lightsail: pacchetto UpdateBucket lightsail: Assistenza UpdateContainer lightsail: UpdateDistribution lightsail: pacchetto UpdateDistribution lightsail: ingresso UpdateDomain lightsail: UpdateInstance MetadataOptions vela leggera: UpdateLoad BalancerAttribute lightsail: Banca dati UpdateRelational lightsail: UpdateRelational DatabaseParameters

Prefisso del servizio	Azioni
log	registri: chiave AssociateKms
	logs: Attività CancelExport
	logs: Attività CreateExport
	registri: CreateLog AnomalyDetector
	registri: Gruppo CreateLog
	registri: Stream CreateLog
	registri: DeleteData ProtectionPolicy
	registri: DeleteDelivery
	registri: destinazione DeleteDelivery
	registri: DeleteDelivery DestinationPolicy
	registri: fonte DeleteDelivery
	registri: DeleteDestination
	registri: Gruppo DeleteLog
	registri: Stream DeleteLog
	log: Filtro DeleteMetric
	log: definizione DeleteQuery
	log: Politica DeleteResource
	log: Politica DeleteRetention
	log: Filtro DeleteSubscription
	registri: politiche DescribeAccount
	registri: DescribeDeliveries

Prefisso del servizio	Azioni
	registri: destinazioni DescribeDelivery
	registri: fonti DescribeDelivery
	registri: DescribeDestinations
	registri: attività DescribeExport
	registri: gruppi DescribeLog
	registri: Streams DescribeLog
	log: filtri DescribeMetric
	registri: DescribeQueries
	registri: definizioni DescribeQuery
	registri: politiche DescribeResource
	registri: filtri DescribeSubscription
	registri: chiave DisassociateKms
	registri: GetData ProtectionPolicy
	registri: GetDelivery
	registri: destinazione GetDelivery
	registri: GetDelivery DestinationPolicy
	registri: fonte GetDelivery
	registri: GetLog GroupFields
	registri: Registra GetLog
	registri: risultati GetQuery
	registri: ListAnomalies

Prefisso del servizio	Azioni
	registri: ListLog AnomalyDetectors
	registri: PutData ProtectionPolicy
	registri: destinazione PutDelivery
	registri: PutDelivery DestinationPolicy
	registri: fonte PutDelivery
	registri: PutDestination
	registri: politica PutDestination
	log: Filtro PutMetric
	log: definizione PutQuery
	log: Politica PutResource
	log: Politica PutRetention
	log: Filtro PutSubscription
	log: Tail StartLive
	tronchi: StartQuery
	registri: StopQuery
	registri: filtro TestMetric

Prefisso del servizio	Azioni
lookoutequipment	attrezzatura di avvistamento: CreateDataset attrezzatura di guardia: Scheduler CreateInference attrezzatura da guardia: CreateLabel attrezzatura di avvistamento: Gruppo CreateLabel attrezzatura di avvistamento: CreateModel attrezzatura di avvistamento: DeleteDataset attrezzatura di guardia: Scheduler DeleteInference attrezzatura da guardia: DeleteLabel attrezzatura di avvistamento: Gruppo DeleteLabel attrezzatura di avvistamento: DeleteModel attrezzatura di avvistamento: politica DeleteResource lookoutequipment: Scheduler DeleteRetraining attrezzatura da guardia: DescribeData IngestionJob attrezzatura di avvistamento: DescribeDataset attrezzatura di guardia: Scheduler DescribeInference lookoutequipment:Describelabel lookoutequipment: Gruppo DescribeLabel attrezzatura di avvistamento: DescribeModel attrezzatura di avvistamento: versione DescribeModel attrezzatura di guardia: politica DescribeResource lookoutequipment: Scheduler DescribeRetraining

Prefisso del servizio	Azioni
	attrezzatura da guardia: ImportDataset attrezzatura di avvistamento: versione ImportModel attrezzatura di avvistamento: ListData IngestionJobs attrezzatura di avvistamento: ListDatasets attrezzatura di avvistamento: eventi ListInference lookoutequipment: Esecuzioni ListInference lookoutequipment: Schedulers ListInference lookoutequipment: Gruppi ListLabel attrezzatura di avvistamento: ListLabels attrezzatura di avvistamento: ListModels attrezzatura di avvistamento: versioni ListModel lookoutequipment: Schedulers ListRetraining lookoutequipment: Statistiche ListSensor lookoutequipment: Politica PutResource attrezzatura di avvistamento: StartData IngestionJob attrezzatura di guardia: Scheduler StartInference lookoutequipment: Scheduler StartRetraining lookoutequipment: Scheduler StopInference lookoutequipment: Scheduler StopRetraining attrezzatura da guardia: UpdateActive ModelVersion attrezzatura di guardia: Scheduler UpdateInference

Prefisso del servizio	Azioni
	lookoutequipment: Gruppo UpdateLabel attrezzatura di avvistamento: UpdateModel attrezzatura di guardia: Scheduler UpdateRetraining

Prefisso del servizio	Azioni
lookoutmetrics	lookoutmetrics: rilevatore ActivateAnomaly metriche di osservazione: BackTest AnomalyDetector metriche di attenzione: CreateAlert lookoutmetrics: rilevatore CreateAnomaly lookoutmetrics: CreateMetric Imposta lookoutmetrics: rilevatore DeactivateAnomaly metriche di osservazione: DeleteAlert lookoutmetrics: rilevatore DeleteAnomaly metriche di osservazione: DescribeAlert metriche di attenzione: DescribeAnomaly DetectionExecutions lookoutmetrics: rilevatore DescribeAnomaly lookoutmetrics: DescribeMetric Imposta metriche di osservazione: DetectMetric SetConfig lookoutmetrics: Gruppo GetAnomaly lookoutmetrics: GetData QualityMetrics metriche di attenzione: GetFeedback lookoutmetrics: dati GetSample metriche di osservazione: ListAlerts lookoutmetrics: rilevatori ListAnomaly lookoutmetrics: ListAnomaly GroupRelated Metriche metriche di osservazione: ListAnomaly GroupSummaries

Prefisso del servizio	Azioni
	lookoutmetrics: Serie ListAnomaly GroupTime
	lookoutmetrics: set ListMetric
	lookoutmetrics: PutFeedback
	metriche di attenzione: UpdateAlert
	lookoutmetrics: rilevatore UpdateAnomaly
	lookoutmetrics: UpdateMetric Imposta

Prefisso del servizio	Azioni
lookoutvision	visione di osservazione: CreateDataset visione d'osservazione: CreateModel visione d'osservazione: CreateProject visione d'osservazione: DeleteDataset visione d'osservazione: DeleteModel visione d'osservazione: DeleteProject visione d'osservazione: DescribeDataset visione d'osservazione: DescribeModel visione d'osservazione: DescribeModel PackagingJob visione d'osservazione: DescribeProject visione d'osservazione: DetectAnomalies lookoutvision: Iscrizioni ListDataset lookoutvision: ListModel PackagingJobs visione d'osservazione: ListModels visione d'osservazione: ListProjects visione d'osservazione: StartModel visione d'osservazione: StartModel PackagingJob visione d'osservazione: StopModel lookoutvision: Iscrizioni UpdateDataset

Prefisso del servizio	Azioni
m2	m2: CancelBatch JobExecution
	m2: CreateApplication
	m2: CreateData SetImport Attività
	m2: CreateDeployment
	m2: CreateEnvironment
	m2: DeleteApplication
	m2: DeleteApplication FromEnvironment
	m2: DeleteEnvironment
	m2: GetApplication
	m2: GetApplication Versione
	m2: GetBatch JobExecution
	m2: GetData SetDetails
	m2: GetData SetImport Attività
	m2: GetDeployment
	m2: GetEnvironment
	m2: GetSigned BluinsightsUrl
	m2: ListApplications
	m2: ListApplication Versioni
	m2: ListBatch JobDefinitions
	m2: ListBatch JobExecutions
	m2: ListBatch JobRestart Punti

Prefisso del servizio	Azioni
	m2: ListData SetImport Storia
	m2: ListData Set
	m2: ListDeployments
	m2: ListEngine Versioni
	m2: ListEnvironments
	m2: StartApplication
	m2: StartBatch Job
	m2: StopApplication
	m2: UpdateApplication
	m2: UpdateEnvironment

Prefisso del servizio	Azioni
managedblockchain	blockchain gestita: CreateAccessor blockchain gestita: CreateMember blockchain gestita: CreateNetwork blockchain gestita: CreateNode blockchain gestita: CreateProposal blockchain gestita: DeleteAccessor blockchain gestita: DeleteMember blockchain gestita: DeleteNode blockchain gestita: GetAccessor blockchain gestita: GetMember blockchain gestita: GetNetwork blockchain gestita: GetNode blockchain gestita: GetProposal blockchain gestita: InvokeRpc PolygonMainnet blockchain gestita: Testnet InvokeRpc PolygonMumbai blockchain gestita: ListAccessors blockchain gestita: ListInvitations blockchain gestita: ListMembers blockchain gestita: ListNetworks blockchain gestita: ListNodes blockchain gestita: ListProposals

Prefisso del servizio	Azioni
	managedblockchain: voti ListProposal blockchain gestita: RejectInvitation blockchain gestita: UpdateMember blockchain gestita: UpdateNode blockchain gestita: proposta VoteOn

Prefisso del servizio	Azioni
mediaconnect	mediaconnect: uscite AddBridge
	mediaconnect: AddBridge Sorgenti
	mediaconnect: AddFlow MediaStreams
	mediaconnect: AddFlow uscite
	mediaconnect: AddFlow Sorgenti
	mediaconnect: AddFlow VpcInterfaces
	connessione multimediale: CreateBridge
	connessione multimediale: CreateFlow
	connessione multimediale: CreateGateway
	connessione multimediale: DeleteBridge
	connessione multimediale: DeleteFlow
	connessione multimediale: DeleteGateway
	mediaconnect: DeregisterGateway Istanza
	mediaconnect: DescribeBridge
	connessione multimediale: DescribeFlow
	connessione multimediale: DescribeFlow SourceMetadata
	connessione multimediale: DescribeGateway
	mediaconnect: DescribeGateway Istanza
	mediaconnect: DescribeOffering
	connessione multimediale: DescribeReservation
	mediaconnect: diritti GrantFlow

Prefisso del servizio	Azioni
	mediaconnect: ListBridges
	connessione multimediale: ListEntitlements
	connessione multimediale: ListFlows
	mediaconnect: Istanze ListGateway
	mediaconnect: ListGateways
	connessione multimediale: ListOfferings
	connessione multimediale: ListReservations
	connessione multimediale: PurchaseOffering
	mediaconnect: uscita RemoveBridge
	mediaconnect: RemoveBridge Fonte
	mediaconnect: RemoveFlow MediaStream
	mediaconnect: uscita RemoveFlow
	mediaconnect: RemoveFlow Fonte
	mediaconnect: RemoveFlow VpcInterface
	mediaconnect: Diritto RevokeFlow
	mediaconnect: StartFlow
	connessione multimediale: StopFlow
	connessione multimediale: UpdateBridge
	mediaconnect: uscita UpdateBridge
	mediaconnect: UpdateBridge Fonte
	mediaconnect: UpdateBridge Stato

Prefisso del servizio	Azioni
	mediaconnect: UpdateFlow
	mediaconnect: Diritto UpdateFlow
	mediaconnect: UpdateFlow MediaStream
	mediaconnect: uscita UpdateFlow
	mediaconnect: UpdateFlow Fonte
	mediaconnect: UpdateGateway Istanza

Prefisso del servizio	Azioni
mediaconvert	mediaconvert: AssociateCertificate conversione di file multimediali: CancelJob conversione di file multimediali: CreateJob mediaconvert: Modello CreateJob mediaconvert: CreatePreset conversione di file multimediali: CreateQueue mediaconvert: Modello DeleteJob mediaconvert: DeletePolicy conversione di file multimediali: DeletePreset conversione di file multimediali: DeleteQueue conversione di file multimediali: DescribeEndpoints conversione di file multimediali: DisassociateCertificate conversione di file multimediali: GetJob mediaconvert: Modello GetJob mediaconvert: GetPolicy conversione di file multimediali: GetPreset conversione di file multimediali: GetQueue conversione di file multimediali: ListJobs mediaconvert: Modelli ListJob mediaconvert: ListPresets conversione di file multimediali: ListQueues

Prefisso del servizio	Azioni
	conversione di file multimediali: PutPolicy mediaconvert: Modello UpdateJob mediaconvert: UpdatePreset conversione di file multimediali: UpdateQueue

Prefisso del servizio	Azioni
medialive	media live: AcceptInput DeviceTransfer
	media in diretta: BatchDelete
	media in diretta: BatchStart
	media in diretta: BatchStop
	medialive: Programma BatchUpdate
	medilive: CancelInput DeviceTransfer
	media in diretta: ClaimDevice
	media in diretta: CreateChannel
	medialive: Modello CreateCloud WatchAlarm
	medilive: CreateCloud WatchAlarm TemplateGroup
	medialive: Modello CreateEvent BridgeRule
	medilive: CreateEvent BridgeRule TemplateGroup
	media in diretta: CreateInput
	media in diretta: CreateInput SecurityGroup
	media in diretta: CreateMultiplex
	medialive: Programma CreateMultiplex
	medialive: Ingresso CreatePartner
	medialive: Mappa CreateSignal
	medilive: DeleteChannel
	medialive: Modello DeleteCloud WatchAlarm
	medilive: DeleteCloud WatchAlarm TemplateGroup

Prefisso del servizio	Azioni
	medialive: Modello DeleteEvent BridgeRule
	medilive: DeleteEvent BridgeRule TemplateGroup
	media in diretta: DeleteInput
	media in diretta: DeleteInput SecurityGroup
	media in diretta: DeleteMultiplex
	medialive: Programma DeleteMultiplex
	medialive: DeleteReservation
	media in diretta: DeleteSchedule
	medialive: Mappa DeleteSignal
	medialive: Configurazione DescribeAccount
	medialive: DescribeChannel
	media in diretta: DescribeInput
	medialive: Dispositivo DescribeInput
	medilive: DescribeInput DeviceThumbnail
	media in diretta: DescribeInput SecurityGroup
	media in diretta: DescribeMultiplex
	medialive: Programma DescribeMultiplex
	medialive: DescribeOffering
	media in diretta: DescribeReservation
	media in diretta: DescribeSchedule
	media in diretta: DescribeThumbnails

Prefisso del servizio	Azioni
	medialive: Modello GetCloud WatchAlarm
	medilive: GetCloud WatchAlarm TemplateGroup
	medialive: Modello GetEvent BridgeRule
	medilive: GetEvent BridgeRule TemplateGroup
	medialive: Mappa GetSignal
	medilive: ListChannels
	media in diretta: ListCloud WatchAlarm TemplateGroups
	medialive: Modelli ListCloud WatchAlarm
	medilive: ListEvent BridgeRule TemplateGroups
	medialive: Modelli ListEvent BridgeRule
	medialive: Dispositivi ListInput
	medilive: ListInput DeviceTransfers
	media in diretta: ListInputs
	media in diretta: ListInput SecurityGroups
	media in diretta: ListMultiplexes
	medialive: Programmi ListMultiplex
	medialive: ListOfferings
	media in diretta: ListReservations
	medialive: mappe ListSignal
	medilive: PurchaseOffering
	medialive: Dispositivo RebootInput

Prefisso del servizio	Azioni
	medilive: RejectInput DeviceTransfer
	medialive: Condutture RestartChannel
	medilive: StartChannel
	media in diretta: StartDelete MonitorDeployment
	medialive: Dispositivo StartInput
	medialive: Finestra StartInput DeviceMaintenance
	medialive: Distribuzione StartMonitor
	medilive: StartMultiplex
	media in diretta: StartUpdate SignalMap
	media in diretta: StopChannel
	medialive: Dispositivo StopInput
	medilive: StopMultiplex
	medialive: Dispositivo TransferInput
	medialive: Configurazione UpdateAccount
	medialive: UpdateChannel
	medialive: Classe UpdateChannel
	medialive: Modello UpdateCloud WatchAlarm
	medilive: UpdateCloud WatchAlarm TemplateGroup
	medialive: Modello UpdateEvent BridgeRule
	medilive: UpdateEvent BridgeRule TemplateGroup
	media in diretta: UpdateInput

Prefisso del servizio	Azioni
	medialive: Dispositivo UpdateInput medilive: UpdateInput SecurityGroup media in diretta: UpdateMultiplex medialive: Programma UpdateMultiplex medialive: UpdateReservation

Prefisso del servizio	Azioni
mediapackage	pacchetto multimediale: ConfigureLogs pacchetto multimediale: CreateChannel mediapackage: Job CreateHarvest mediapackage: Endpoint CreateOrigin pacchetto multimediale: DeleteChannel pacchetto multimediale: Endpoint DeleteOrigin pacchetto multimediale: DescribeChannel mediapackage: Job DescribeHarvest mediapackage: Endpoint DescribeOrigin pacchetto multimediale: ListChannels mediapackage: Offerte di lavoro ListHarvest mediapackage: Endpoints ListOrigin pacchetto multimediale: ListTags ForResource mediapackage: credenziali RotateChannel pacchetto multimediale: RotateIngest EndpointCredentials pacchetto multimediale: TagResource pacchetto multimediale: UntagResource pacchetto multimediale: UpdateChannel pacchetto multimediale: Endpoint UpdateOrigin

Prefisso del servizio	Azioni
mediapackage-vod	pacchetto media-vod: ConfigureLogs pacchetto media-vod: CreateAsset mediapackage-vod: Configurazione CreatePackaging mediapackage-vod: Gruppo CreatePackaging pacchetto media-vod: DeleteAsset mediapackage-vod: Configurazione DeletePackaging mediapackage-vod: Gruppo DeletePackaging pacchetto media-vod: DescribeAsset mediapackage-vod: Configurazione DescribePackaging mediapackage-vod: Gruppo DescribePackaging pacchetto media-vod: ListAssets mediapackage-vod: Configurazioni ListPackaging mediapackage-vod: Gruppi ListPackaging pacchetto media-vod: ListTags ForResource pacchetto media-vod: TagResource pacchetto media-vod: UntagResource mediapackage-vod: Gruppo UpdatePackaging

Prefisso del servizio	Azioni
mediastore	mediastore: CreateContainer mediastore: DeleteContainer mediastore: Politica DeleteContainer mediastore: Politica DeleteCors mediastore: Politica DeleteLifecycle mediastore: Politica DeleteMetric mediastore: DescribeContainer mediastore: Politica GetContainer mediastore: Politica GetCors mediastore: Politica GetLifecycle mediastore: Politica GetMetric mediastore: ListContainers mediastore: Politica PutContainer mediastore: Politica PutCors mediastore: Politica PutLifecycle mediastore: Politica PutMetric mediastore: Registrazione StartAccess mediastore: Registrazione StopAccess

Prefisso del servizio	Azioni
mediatailor	mediatailor: Configurazione ConfigureLogs ForPlayback
	mediatailor: CreateChannel
	mediatailor: Fonte CreateLive
	mediatailor: Pianificazione CreatePrefetch
	mediatailor: CreateProgram
	mediatailor: Posizione CreateSource
	mediatailor: Fonte CreateVod
	mediatailor: DeleteChannel
	mediatailor: Politica DeleteChannel
	mediatailor: Fonte DeleteLive
	mediatailor: Configurazione DeletePlayback
	mediatailor: Pianifica DeletePrefetch
	mediatailor: DeleteProgram
	mediatailor: Posizione DeleteSource
	mediatailor: Fonte DeleteVod
	mediatailor: DescribeChannel
	mediatailor: Fonte DescribeLive
	mediatailor: DescribeProgram
	mediatailor: Posizione DescribeSource
	mediatailor: Fonte DescribeVod
	mediatailor: Politica GetChannel

Prefisso del servizio	Azioni
	mediatailor: Programma GetChannel
	mediatailor: Configurazione GetPlayback
	mediatailor: Pianifica GetPrefetch
	mediatailor: ListAlerts
	mediatailor: ListChannels
	mediatailor: Fonti ListLive
	mediatailor: Configurazioni ListPlayback
	mediatailor: ListPrefetch Pianificazioni
	mediatailor: Sedi ListSource
	mediatailor: Fonti ListVod
	mediatailor: Politica PutChannel
	mediatailor: Configurazione PutPlayback
	mediatailor: StartChannel
	mediatailor: StopChannel
	mediatailor: UpdateChannel
	mediatailor: Fonte UpdateLive
	mediatailor: UpdateProgram
	mediatailor: Posizione UpdateSource
	mediatailor: Fonte UpdateVod

Prefisso del servizio	Azioni
memorydb	memorydb: Cluster BatchUpdate memorydb: CopySnapshot db di memoria: CreateAcl db di memoria: CreateCluster memorydb: Gruppo CreateParameter memorydb: CreateSnapshot memorydb: Gruppo CreateSubnet memorydb: CreateUser db di memoria: DeleteAcl db di memoria: DeleteCluster memorydb: Gruppo DeleteParameter memorydb: DeleteSnapshot memorydb: Gruppo DeleteSubnet memorydb: DeleteUser db di memoria: DescribeAcls db di memoria: DescribeClusters memorydb: Versioni DescribeEngine memorydb: DescribeEvents memorydb: Gruppi DescribeParameter memorydb: DescribeParameters memorydb: nodi DescribeReserved

Prefisso del servizio	Azioni
	memorydb: DescribeReserved NodesOfferings memorydb: Aggiornamenti DescribeService memorydb: DescribeSnapshots memorydb: Gruppi DescribeSubnet memorydb: DescribeUsers db di memoria: FailoverShard memorydb: Aggiornamenti ListAllowed NodeType memorydb: PurchaseReserved NodesOffering memorydb: Gruppo ResetParameter memorydb: UpdateAcl db di memoria: UpdateCluster memorydb: Gruppo UpdateParameter memorydb: Gruppo UpdateSubnet memorydb: UpdateUser

Prefisso del servizio	Azioni
mgh	mgh: AssociateCreated Artefatto
	mgh: Risorsa AssociateDiscovered
	mgh: CreateHome RegionControl
	mg: CreateProgress UpdateStream
	mg: DeleteHome RegionControl
	mg: DeleteProgress UpdateStream
	mgh: Stato DescribeApplication
	mgh: DescribeHome RegionControls
	mgh: Attività DescribeMigration
	mgh: DisassociateCreated Artefatto
	mgh: Risorsa DisassociateDiscovered
	mgh: Regione GetHome
	mgh: Attività ImportMigration
	mgh: Stati ListApplication
	mgh: Artefatti ListCreated
	mgh: Risorse ListDiscovered
	mgh: Attività ListMigration
	mgh: ListProgress UpdateStreams
	mgh: Stato NotifyApplication
	mgh: NotifyMigration TaskState
	mgh: Attributi PutResource

Prefisso del servizio	Azioni
mgn	mgn: ArchiveApplication mgn: ArchiveWave mgn: AssociateApplications mgn: Server AssociateSource mgn: Stato ChangeServer LifeCycle mgn: CreateApplication mgn: CreateConnector mgn: CreateLaunch ConfigurationTemplate mgn: CreateReplication ConfigurationTemplate mgn: CreateWave mgn: DeleteApplication mgn: DeleteConnector mgn: DeleteJob mgn: DeleteLaunch ConfigurationTemplate mgn: DeleteReplication ConfigurationTemplate mgn: Server DeleteSource mgn: Cliente DeleteVcenter mgn: DeleteWave mgn: DescribeJob LogItems mgn: DescribeJobs mgn: DescribeLaunch ConfigurationTemplates

Prefisso del servizio	Azioni
	mgn: DescribeReplication ConfigurationTemplates mgn: Clienti DescribeVcenter mgn: DisassociateApplications mgn: Server DisassociateSource mgn: Servizio DisconnectFrom mgn: FinalizeCutover mgn: Configurazione GetReplication mgn: InitializeService mgn: ListConnectors mgn: Errori ListExport mgn: ListExports mgn: Errori ListImport mgn: ListImports mgn: Conti ListManaged mgn: ListSource ServerActions mgn: Azioni ListTemplate mgn: archiviato MarkAs mgn: PauseReplication mgn: PutSource ServerAction mgn: Azione PutTemplate mgn: RemoveSource ServerAction

Prefisso del servizio	Azioni
	mgn: Azione RemoveTemplate mgn: ResumeReplication mgn: Replica RetryData mgn: StartCutover mgn: StartExport mgn: StartImport mgn: StartReplication mgn: StartTest mgn: StopReplication mgn: Istanze TerminateTarget mgn: UnarchiveApplication mgn: UnarchiveWave mgn: UpdateApplication mgn: UpdateConnector mgn: UpdateLaunch ConfigurationTemplate mgn: Configurazione UpdateReplication mgn: UpdateReplication ConfigurationTemplate mgn: Server UpdateSource mgn: Tipo UpdateSource ServerReplication mgn: UpdateWave

Prefisso del servizio	Azioni
migrationhub-strategy	migrationhub-strategy: modello GetAnti
	migrationhub-strategy: GetApplication ComponentDetails
	migrationhub-strategy: GetApplication ComponentStrategies
	migrationhub-strategy: GetAssessment
	migrationhub-strategy: GetImport FileTask
	migrationhub-strategy: GetLatest AssessmentId
	migrationhub-strategy: GetMessage
	migrationhub-strategy: Preferenze GetPortfolio
	migrationhub-strategy: Riepilogo GetPortfolio
	migrationhub-strategy: GetRecommendation ReportDetails
	migrationhub-strategy: Dettagli GetServer
	migrationhub-strategy: strategie GetServer
	migrationhub-strategy: server ListAnalyzable
	migrationhub-strategy: modelli ListAnti
	migrationhub-strategy: componenti ListApplication
	migrationhub-strategy: ListCollectors
	migrationhub-strategy: ListImport FileTask
	migrationhub-strategy: artefatti ListJar
	migrationhub-strategy: ListServers
	migrationhub-strategy: Preferenze PutPortfolio
	migrationhub-strategy: RegisterCollector

Prefisso del servizio	Azioni
	migrationhub-strategy: SendMessage
	migrationhub-strategy: StartAssessment
	migrationhub-strategy: StartImport FileTask
	migrationhub-strategy: StartRecommendation ReportGeneration
	migrationhub-strategy: StopAssessment
	migrationhub-strategy: UpdateApplication ComponentConfig
	migrationhub-strategy: configurazione UpdateCollector
	migrationhub-strategy: Config UpdateServer

Prefisso del servizio	Azioni
mobiletargeting	targeting mobile: CreateApp targeting mobile: CreateCampaign mobiletargeting: modello CreateEmail mobiletargeting: Job CreateExport mobiletargeting: Job CreateImport targeting mobile: CreateIn AppTemplate targeting mobile: CreateJourney mobiletargeting: modello CreatePush mobiletargeting: configurazione CreateRecommender targeting mobile: CreateSegment mobiletargeting: modello CreateSms mobiletargeting: modello CreateVoice mobiletargeting: canale DeleteAdm mobiletargeting: canale DeleteApns targeting mobile: DeleteApns SandboxChannel targeting mobile: DeleteApns VoipChannel mobiletargeting: canale DeleteApns VoipSandbox targeting mobile: DeleteApp mobiletargeting: canale DeleteBaidu targeting mobile: DeleteCampaign mobiletargeting: canale DeleteEmail

Prefisso del servizio	Azioni
	mobiletargeting: modello DeleteEmail targeting mobile: DeleteEndpoint targeting mobile: Stream DeleteEvent mobiletargeting: canale DeleteGcm targeting mobile: DeleteIn AppTemplate targeting mobile: DeleteJourney mobiletargeting: modello DeletePush mobiletargeting: configurazione DeleteRecommender targeting mobile: DeleteSegment mobiletargeting: canale DeleteSms mobiletargeting: modello DeleteSms mobiletargeting: endpoint DeleteUser mobiletargetingDeleteVoice: canale mobiletargeting: modello DeleteVoice mobiletargeting: canale GetAdm mobiletargeting: canale GetApns targeting mobile: GetApns SandboxChannel targeting mobile: GetApns VoipChannel mobiletargeting: canale GetApns VoipSandbox targeting mobile: GetApp targeting mobile: Kpi GetApplication DateRange

Prefisso del servizio	Azioni
	mobiletargeting: GetApplication impostazioni targeting mobile: GetApps mobiletargeting: canale GetBaidu targeting mobile: GetCampaign mobiletargeting: attività GetCampaign mobiletargeting: Kpi GetCampaign DateRange targeting mobile: GetCampaigns mobiletargeting: versione GetCampaign mobiletargeting: versioni GetCampaign targeting mobile: GetChannels mobiletargeting: canale GetEmail mobiletargeting: modello GetEmail targeting mobile: GetEndpoint targeting mobile: Stream GetEvent mobiletargeting: Job GetExport mobiletargeting: Offerte di lavoro GetExport mobiletargeting: canale GetGcm mobiletargeting: Job GetImport mobiletargeting: Offerte di lavoro GetImport targeting mobile: GetIn AppMessages targeting mobile: GetIn AppTemplate

Prefisso del servizio	Azioni
	targeting mobile: GetJourney targeting mobile: Kpi GetJourney DateRange mobiletargeting: GetJourney ExecutionActivity metriche targeting mobile: GetJourney ExecutionMetrics targeting mobile: GetJourney RunExecution ActivityMetrics targeting mobile: metriche GetJourney RunExecution mobiletargeting: GetJourney funziona mobiletargeting: modello GetPush mobiletargeting: configurazione GetRecommender mobiletargeting: configurazioni GetRecommender targeting mobile: GetSegment targeting mobile: GetSegment ExportJobs targeting mobile: GetSegment ImportJobs targeting mobile: GetSegments mobiletargeting: versione GetSegment mobiletargeting: versioni GetSegment mobiletargeting: canale GetSms mobiletargeting: modello GetSms mobiletargeting: endpoint GetUser mobiletargetingGetVoice: canale mobiletargeting: modello GetVoice

Prefisso del servizio	Azioni
	targeting mobile: ListJourneys targeting mobile: ListTemplates mobiletargeting: versioni ListTemplate mobiletargeting: convalida PhoneNumber mobiletargeting: PutEvent Stream targeting mobile: RemoveAttributes mobiletargeting: canale UpdateAdm mobiletargeting: canale UpdateApns targeting mobile: UpdateApns SandboxChannel targeting mobile: UpdateApns VoipChannel mobiletargeting: canale UpdateApns VoipSandbox mobiletargeting: impostazioni UpdateApplication mobiletargeting: canale UpdateBaidu targeting mobile: UpdateCampaign mobiletargeting: canale UpdateEmail mobiletargeting: modello UpdateEmail targeting mobile: UpdateEndpoint targeting mobile: Batch UpdateEndpoints mobiletargeting: canale UpdateGcm targeting mobile: UpdateIn AppTemplate targeting mobile: UpdateJourney

Prefisso del servizio	Azioni
	mobiletargeting: Stato UpdateJourney mobiletargeting: modello UpdatePush mobiletargeting: configurazione UpdateRecommender targeting mobile: UpdateSegment mobiletargeting: canale UpdateSms mobiletargeting: modello UpdateSms targeting mobile: UpdateTemplate ActiveVersion mobiletargeting: canale UpdateVoice mobiletargeting: modello UpdateVoice mobiletargeting:VerifyOTPMessage

Prefisso del servizio	Azioni
mq	mq: CreateBroker mq: CreateConfiguration mq: CreateUser mq: DeleteBroker mq: DeleteUser mq: DescribeBroker mq: DescribeBroker EngineTypes mq: DescribeBroker InstanceOptions mq: DescribeConfiguration mq: DescribeConfiguration Revisione mq: DescribeUser mq: ListBrokers mq: ListConfiguration Revisioni mq: ListConfigurations mq: ListUsers mq:Promote mq: RebootBroker mq: UpdateBroker mq: UpdateConfiguration mq: UpdateUser

Prefisso del servizio	Azioni
networkmanager	gestore di rete: AcceptAttachment gestore di rete: Peer AssociateConnect gestore di rete: Gateway AssociateCustomer gestore di rete: AssociateLink gestore di rete: Peer AssociateTransit GatewayConnect networkmanager: Allegato CreateConnect gestore di rete: CreateConnection gestore di rete: Peer CreateConnect gestore di rete: rete CreateCore gestore di rete: CreateDevice gestore di rete: rete CreateGlobal gestore di rete: CreateLink gestore di rete: CreateSite gestore di rete: CreateSite ToSite VpnAttachment gestore di rete: CreateTransit GatewayPeering gestore di rete: CreateTransit GatewayRoute TableAttachment networkmanager: Allegato CreateVpc gestore di rete: DeleteAttachment gestore di rete: DeleteConnection gestore di rete: Peer DeleteConnect gestore di rete: rete DeleteCore

Prefisso del servizio	Azioni
	networkmanager: versione DeleteCore NetworkPolicy gestore di rete: DeleteDevice gestore di rete: rete DeleteGlobal gestore di rete: DeleteLink gestore di rete: DeletePeering networkmanager: politica DeleteResource gestore di rete: DeleteSite gestore di rete: Gateway DeregisterTransit gestore di rete: reti DescribeGlobal gestore di rete: Peer DisassociateConnect gestore di rete: Gateway DisassociateCustomer gestore di rete: DisassociateLink gestore di rete: Peer DisassociateTransit GatewayConnect gestore di rete: Imposta ExecuteCore NetworkChange networkmanager: allegato GetConnect gestore di rete: GetConnections gestore di rete: Peer GetConnect gestore di rete: GetConnect PeerAssociations gestore di rete: rete GetCore networkmanager: Eventi GetCore NetworkChange gestore di rete: Imposta GetCore NetworkChange

Prefisso del servizio	Azioni
	gestore di rete: GetCore NetworkPolicy gestore di rete: GetCustomer GatewayAssociations gestore di rete: GetDevices networkmanager: Associazioni GetLink gestore di rete: GetLinks gestore di rete: GetNetwork ResourceCounts gestore di rete: GetNetwork ResourceRelationships networkmanager: Risorse GetNetwork networkmanager: Percorsi GetNetwork networkmanager: telemetria GetNetwork networkmanagerGetResource: politica networkmanager: analisi GetRoute gestore di rete: GetSites gestore di rete: GetSite ToSite VpnAttachment gestore di rete: GetTransit GatewayConnect PeerAssociations gestore di rete: GetTransit GatewayPeering gestore di rete: GetTransit GatewayRegistrations gestore di rete: GetTransit GatewayRoute TableAttachment networkmanager: Allegato GetVpc gestore di rete: ListAttachments gestore di rete: colleghi ListConnect

Prefisso del servizio	Azioni
	networkmanager: versioni ListCore NetworkPolicy networkmanager: reti ListCore gestore di rete: stato ListOrganization ServiceAccess gestore di rete: ListPeerings gestore di rete: PutCore NetworkPolicy networkmanager: politica PutResource gestore di rete: Gateway RegisterTransit gestore di rete: RejectAttachment networkmanager: versione RestoreCore NetworkPolicy networkmanager: Aggiornamento StartOrganization ServiceAccess networkmanager: analisi StartRoute gestore di rete: UpdateConnection gestore di rete: rete UpdateCore gestore di rete: UpdateDevice gestore di rete: rete UpdateGlobal gestore di rete: UpdateLink gestore di rete: UpdateNetwork ResourceMetadata gestore di rete: UpdateSite networkmanager: Allegato UpdateVpc

Prefisso del servizio	Azioni
nimble	agile: AcceptEulas nimble: Profilo CreateLaunch nimble: Immagine CreateStreaming nimble: Sessione CreateStreaming agile: CreateStreaming SessionStream agile: CreateStudio agile: Componente CreateStudio nimble: Profilo DeleteLaunch agile: DeleteLaunch ProfileMember nimble: Immagine DeleteStreaming nimble: Sessione DeleteStreaming agile: DeleteStudio agile: Componente DeleteStudio nimble: Membro DeleteStudio agile: GetEula agile: GetLaunch ProfileDetails nimble: Immagine GetStreaming nimble: Sessione GetStreaming agile: GetStreaming SessionBackup agile: GetStreaming SessionStream agile: GetStudio

Prefisso del servizio	Azioni
	agile: Componente GetStudio
	nimble: Membro GetStudio
	agile: ListEulas
	agile: ListLaunch ProfileMembers
	nimble: Profili ListLaunch
	nimble: Immagini ListStreaming
	agile: ListStreaming SessionBackups
	nimble: sessioni ListStreaming
	nimble: Componenti ListStudio
	nimble: Membri ListStudio
	agile: ListStudios
	agile: PutLaunch ProfileMembers
	nimble: Membri PutStudio
	nimble: Sessione StartStreaming
	agile: SSO StartStudio ConfigurationRepair
	nimble: Sessione StopStreaming
	nimble: Profilo UpdateLaunch
	agile: UpdateLaunch ProfileMember
	nimble: Immagine UpdateStreaming
	agile: UpdateStudio
	agile: Componente UpdateStudio

Prefisso del servizio	Azioni
omics	fumetti: Carica AbortMultipart ReadSet Fumetti: BatchDelete ReadSet fumetti: CancelAnnotation ImportJob fumetti: CancelRun fumetti: CancelVariant ImportJob fumetti: carica CompleteMultipart ReadSet Fumetti: Store CreateAnnotation Fumetti: Carica CreateMultipart ReadSet Fumetti: Store CreateReference Fumetti: Gruppo CreateRun Fumetti: Store CreateSequence Fumetti: Store CreateVariant Fumetti: CreateWorkflow fumetti: Store DeleteAnnotation Fumetti: DeleteReference fumetti: Store DeleteReference Fumetti: DeleteRun fumetti: Gruppo DeleteRun Fumetti: Store DeleteSequence Fumetti: Store DeleteVariant Fumetti: DeleteWorkflow

Prefisso del servizio	Azioni
	fumetti: GetAnnotation ImportJob fumetti: Store GetAnnotation Fumetti: Set GetRead Fumetti: Job GetRead SetActivation Fumetti: Job GetRead SetExport Fumetti: Job GetRead SetImport Fumetti: GetRead SetMetadata fumetti: GetReference fumetti: GetReference ImportJob fumetti: metadati GetReference Fumetti: Store GetReference Fumetti: GetRun fumetti: Gruppo GetRun Fumetti: Attività GetRun Fumetti: Store GetSequence Fumetti: GetVariant ImportJob fumetti: Store GetVariant Fumetti: GetWorkflow fumetti: ListAnnotation ImportJobs Fumetti: Storie ListAnnotation Fumetti: caricamenti ListMultipart ReadSet

Prefisso del servizio	Azioni
	Fumetti: Offerte di lavoro ListRead SetActivation Fumetti: Offerte di lavoro ListRead SetExport Fumetti: Offerte di lavoro ListRead SetImport Fumetti: set ListRead Fumetti: parti ListRead SetUpload fumetti: ListReference ImportJobs fumetti: ListReferences Fumetti: Storie ListReference Fumetti: gruppi ListRun Fumetti: ListRuns fumetti: compiti ListRun Fumetti: storie ListSequence Fumetti: ListVariant ImportJobs Fumetti: Storie ListVariant Fumetti: ListWorkflows fumetti: StartAnnotation ImportJob Fumetti: Job StartRead SetActivation Fumetti: Job StartRead SetExport Fumetti: Job StartRead SetImport Fumetti: StartReference ImportJob fumetti: StartRun

Prefisso del servizio	Azioni
	fumetti: StartVariant ImportJob fumetti: Store UpdateAnnotation Fumetti: Gruppo UpdateRun Fumetti: Store UpdateVariant Fumetti: UpdateWorkflow fumetti: UploadRead SetPart

Prefisso del servizio	Azioni
opsworks	Ops funziona: AssignInstance
	opsworks: AssignVolume
	opsworks:Ip AssociateElastic
	opsworks: AttachElastic LoadBalancer
	opsworks: CloneStack
	opsworks: CreateApp
	opsworks: CreateDeployment
	opsworks: CreateInstance
	opsworks: CreateLayer
	opsworks: CreateStack
	opsworks: Profilo CreateUser
	opsworks: DeleteApp
	opsworks: DeleteInstance
	opsworks: DeleteLayer
	opsworks: DeleteStack
	opsworks: Profilo DeleteUser
	opsworks: Cluster DeregisterEcs
	opsworks: Ip DeregisterElastic
	opsworks: DeregisterInstance
	opsworks: DeregisterRds DbInstance
	opsworks: DeregisterVolume

Prefisso del servizio	Azioni
	opsworks: Versioni DescribeAgent
	opsworks: DescribeApps
	opsworks: DescribeCommands
	opsworks: DescribeDeployments
	opsworks: cluster DescribeEcs
	opsworks: Suggerimenti DescribeElastic
	opsworks: DescribeElastic LoadBalancers
	opsworks: DescribeInstances
	opsworks: DescribeLayers
	opsworks: Scalabilità DescribeLoad BasedAuto
	opsworks: DescribeMy UserProfile
	opsworks: Sistemi DescribeOperating
	opsworks: DescribePermissions
	opsworks: matrici DescribeRaid
	opsworks: DescribeRds DbInstances
	opsworks: errori DescribeService
	opsworks: DescribeStack ProvisioningParameters
	opsworks: DescribeStacks
	opsworks: Riepilogo DescribeStack
	opsworks: Scalabilità DescribeTime BasedAuto
	opsworks: Profili DescribeUser

Prefisso del servizio	Azioni
	opsworks: DescribeVolumes
	opsworks: DetachElastic LoadBalancer
	opsworks:Ip DisassociateElastic
	opsworks: suggerimento GetHostname
	opsworks: GrantAccess
	opsworks: RebootInstance
	opsworks: Cluster RegisterEcs
	opsworks: Ip RegisterElastic
	opsworks: RegisterInstance
	opsworks: RegisterRds DbInstance
	opsworks: RegisterVolume
	opsworks: Scalabilità SetLoad BasedAuto
	opsworks: SetPermission
	opsworks: Scalabilità SetTime BasedAuto
	opsworks: StartInstance
	opsworks: StartStack
	opsworks: StopInstance
	opsworks: StopStack
	opsworks: UnassignInstance
	opsworks: UnassignVolume
	opsworks: UpdateApp

Prefisso del servizio	Azioni
	opsworks:Ip UpdateElastic
	opsworks: UpdateInstance
	opsworks: UpdateLayer
	opsworks: UpdateMy UserProfile
	opsworks: UpdateRds DbInstance
	opsworks: UpdateStack
	opsworks: Profilo UpdateUser
	opsworks: UpdateVolume

Prefisso del servizio	Azioni
opsworks-cm	opsworks-cm: AssociateNode opsworks-cm: CreateBackup opsworks-cm: CreateServer opsworks-cm: DeleteBackup opsworks-cm: DeleteServer opsworks-cm: Attributi DescribeAccount opsworks-cm: DescribeBackups opsworks-cm: DescribeEvents opsworks-cm: DescribeNode AssociationStatus opsworks-cm: DescribeServers opsworks-cm: DisassociateNode opsworks-cm: ExportServer EngineAttribute opsworks-cm: RestoreServer opsworks-cm: StartMaintenance opsworks-cm: UpdateServer opsworks-cm: UpdateServer EngineAttributes

Prefisso del servizio	Azioni
organizations	organizzazioni: AcceptHandshake
	organizzazioni: AttachPolicy
	organizzazioni: CancelHandshake
	organizzazioni: CloseAccount
	organizzazioni: CreateAccount
	organizzazioni: CreateGov CloudAccount
	organizzazioni: CreateOrganization
	organizzazioni: CreateOrganizational Unità
	organizzazioni: CreatePolicy
	organizzazioni: DeclineHandshake
	organizzazioni: DeleteOrganization
	organizzazioni: DeleteOrganizational Unità
	organizzazioni: DeletePolicy
	organizzazioni: DeleteResource politica
	organizzazioni: DeregisterDelegated amministratore
	organizzazioni: DescribeAccount
	organizzazioni: DescribeCreate AccountStatus
	organizzazioni: DescribeEffective politica
	organizzazioni: DescribeHandshake
	organizzazioni: DescribeOrganization
	organizzazioni: DescribeOrganizational Unità

Prefisso del servizio	Azioni
	organizzazioni: DescribePolicy
	organizzazioni: DescribeResource politica
	organizzazioni: DetachPolicy
	organizzazioni: disabilita AWSServiceAccess
	organizzazioni: tipo DisablePolicy
	organizzazioni: EnableAll caratteristiche
	Organizzazioni: abilita AWSServiceAccess
	organizzazioni: tipo EnablePolicy
	organizzazioni: InviteAccount ToOrganization
	organizzazioni: LeaveOrganization
	organizzazioni: ListAccounts
	organizzazioni: ListAccounts ForParent
	organizzazioni: elenco AWSServiceAccessForOrganization
	organizzazioni: ListChildren
	organizzazioni: ListCreate AccountStatus
	organizzazioni: ListDelegated amministratori
	organizzazioni: Account ListDelegated ServicesFor
	organizzazioni: ListHandshakes ForAccount
	organizzazioni: ListHandshakes ForOrganization
	organizzazioni: ListOrganizational UnitsFor Genitore
	organizzazioni: ListParents

Prefisso del servizio	Azioni
	organizzazioni: ListPolicies
	organizzazioni: ListPolicies ForTarget
	organizzazioni: ListRoots
	organizzazioni: ListTargets ForPolicy
	organizzazioni: MoveAccount
	organizzazioni: PutResource politica
	organizzazioni: RegisterDelegated amministratore
	organizzazioni: RemoveAccount FromOrganization
	organizzazioni: UpdateOrganizational Unità
	organizzazioni: UpdatePolicy

Prefisso del servizio	Azioni
outposts	avamposti: CancelCapacity Task avamposti: CancelOrder avamposti: CreateOrder avamposti: CreateOutpost avamposti: CreatePrivate ConnectivityConfig avamposti: CreateSite avamposti: DeleteOutpost avamposti: DeleteSite avamposti: Task GetCapacity avamposti: Oggetto GetCatalog avamposti: GetConnection avamposti: GetOrder avamposti: GetOutpost avamposti: GetOutpost InstanceTypes avamposti: tipi GetOutpost SupportedInstance avamposti: GetPrivate ConnectivityConfig avamposti: GetSite avamposti: Indirizzo GetSite avamposti: ListAssets avamposti: compiti ListCapacity avamposti: Oggetti ListCatalog

Prefisso del servizio	Azioni
	avamposti: ListOrders avamposti: ListOutposts avamposti: ListSites avamposti: Task StartCapacity avamposti: StartConnection avamposti: UpdateOutpost avamposti: UpdateSite avamposti: Indirizzo UpdateSite avamposti: Proprietà UpdateSite RackPhysical

Prefisso del servizio	Azioni
panorama	panorama: CreateApplication Istanza
	panorama: CreateJob ForDevices
	panorama: CreateNode FromTemplate Job
	panorama: CreatePackage
	panorama: CreatePackage ImportJob
	panorama: DeleteDevice
	panorama: DeletePackage
	panorama: DeregisterPackage versione
	panorama: DescribeApplication Istanza
	panorama: DescribeApplication InstanceDetails
	panorama: DescribeDevice
	panorama: DescribeDevice Job
	panorama: DescribeNode
	panorama: DescribeNode FromTemplate Job
	panorama: DescribePackage
	panorama: DescribePackage ImportJob
	panorama: DescribePackage versione
	panorama: ListApplication InstanceDependencies
	panorama: ListApplication InstanceNode Istanze
	panorama: Istanze ListApplication
	panorama: ListDevices

Prefisso del servizio	Azioni
	panorama: ListDevices Offerte di lavoro panorama: ListNode FromTemplate Offerte di lavoro panorama: ListNodes panorama: ListPackage ImportJobs panorama: ListPackages panorama: ProvisionDevice panorama: RegisterPackage versione panorama: RemoveApplication Istanza panorama: SignalApplication InstanceNode Istanze panorama: UpdateDevice Metadati
pi	pip: CreatePerformance AnalysisReport perno: DeletePerformance AnalysisReport pi: DescribeDimension chiavi pi: GetDimension KeyDetails perno: GetPerformance AnalysisReport pi: GetResource Metadati pi: Metriche GetResource pi: ListAvailable ResourceDimensions perno: ListAvailable ResourceMetrics perno: ListPerformance AnalysisReports

Prefisso del servizio	Azioni
pipes	tubi: CreatePipe tubi: DeletePipe tubi: DescribePipe tubi: ListPipes tubi: StartPipe tubi: StopPipe tubi: UpdatePipe
polly	polly: DeleteLexicon polly: DescribeVoices polly: GetLexicon polly: GetSpeech SynthesisTask polly: ListLexicons polly: ListSpeech SynthesisTasks polly: PutLexicon polly: StartSpeech SynthesisTask polly: SynthesizeSpeech

Prefisso del servizio	Azioni
profilo	profilo: Key AddProfile
	profilo: CreateCalculated AttributeDefinition
	profilo: CreateDomain
	profilo: CreateEvent Stream
	profilo: CreateProfile
	profilo: DeleteCalculated AttributeDefinition
	profilo: DeleteDomain
	profilo: DeleteEvent Stream
	profilo: DeleteIntegration
	profilo: DeleteProfile
	profilo: DeleteProfile Key
	profilo: DeleteProfile oggetto
	profilo: DeleteProfile ObjectType
	profilo: DeleteWorkflow
	profilo: DetectProfile ObjectType
	profilo: GetAuto MergingPreview
	profilo: GetCalculated AttributeDefinition
	profilo: GetCalculated AttributeFor Profilo
	profilo: GetDomain
	profilo: GetEvent Stream
	profilo: GetIdentity ResolutionJob

Prefisso del servizio	Azioni
	profilo: GetIntegration
	profilo: GetMatches
	profilo: GetProfile ObjectType
	profilo: GetProfile ObjectType modello
	profilo: GetSimilar Profili
	profilo: GetWorkflow
	profilo: GetWorkflow Steps
	profilo: ListAccount Integrazioni
	profilo: ListCalculated AttributeDefinitions
	profilo: ListCalculated AttributesFor Profilo
	profilo: ListDomains
	profilo: ListEvent Streams
	profilo: ListIdentity ResolutionJobs
	profilo: ListIntegrations
	profilo: ListProfile Oggetti
	profilo: ListProfile ObjectTypes
	profilo: ListProfile ObjectType modelli
	profilo: ListRule BasedMatches
	profilo: ListWorkflows
	profilo: MergeProfiles
	profilo: PutIntegration

Prefisso del servizio	Azioni
	profilo: PutProfile oggetto profilo: PutProfile ObjectType profilo: SearchProfiles profilo: UpdateCalculated AttributeDefinition profilo: UpdateDomain profilo: UpdateProfile

Prefisso del servizio	Azioni
qldb	qldb: CancelJournal KinesisStream qldb: CreateLedger qldb: DeleteLedger qldb: DescribeJournal KinesisStream qldb: S3Export DescribeJournal qldb: DescribeLedger qldb: ToS3 ExportJournal qldb: GetBlock qldb: GetDigest qldb: GetRevision qldb: ListJournal KinesisStreams ForLedger qldb: S3Exports ListJournal qldb: registro S3 ListJournal ExportsFor qldb: ListLedgers qldb: StreamJournal ToKinesis qldb: UpdateLedger qldb: UpdateLedger PermissionsMode

Prefisso del servizio	Azioni
ram	ram: AcceptResource ShareInvitation
	ram: AssociateResource Condividi
	ram: AssociateResource SharePermission
	ram: CreatePermission
	ram: CreatePermission versione
	ram: CreateResource Condividi
	ram: DeletePermission
	ram: DeletePermission versione
	ram: DeleteResource Condividi
	ram: DisassociateResource Condividi
	ram: DisassociateResource SharePermission
	ram: EnableSharing WithAws Organizzazione
	ram: GetPermission
	ram: GetResource politiche
	ram: GetResource ShareAssociations
	ram: GetResource ShareInvitations
	ram: GetResource condivisioni
	ram: ListPending InvitationResources
	ram: ListPermission Associazioni
	ram: ListPermissions
	ram: ListPermission Versioni

Prefisso del servizio	Azioni
	ram: ListPrincipals ram: ListReplace PermissionAssociations lavoro ram: ListResources ram: ListResource SharePermissions ram: ListResource tipi ram: PromotePermission CreatedFrom politica ram: PromoteResource ShareCreated FromPolicy ram: RejectResource ShareInvitation ram: ReplacePermission Associazioni ram: SetDefault PermissionVersion ram: UpdateResource Condividi
rbin	rbin: CreateRule rubino: DeleteRule rubino: GetRule rubino: ListRules rubino: LockRule rubino: UnlockRule rubino: UpdateRule

Prefisso del servizio	Azioni
rds	rds: toDbCluster AddRole
	rds: toDbInstance AddRole
	rds: Abbonamento AddSource IdentifierTo
	rds: ApplyPending MaintenanceAction
	SecurityGroupRDS: ingresso B autorizzato
	rds:BacktrackDBCluster
	rds: Attività CancelExport
	ClusterParameterrds:Gruppo CopyDB
	RDS: copyDB ClusterSnapshot
	RDS: copyDB ParameterGroup
	rds:CopyDBSnapshot
	rds: Gruppo CopyOption
	rds: DB CreateCustom EngineVersion
	ClusterParameterRDS: crea un gruppo DB
	RDS: creato DB ClusterSnapshot
	RDS: creato DB ParameterGroup
	rds:CreateDBProxy
	RDS: creato DB ProxyEndpoint
	RDS: creato DB SecurityGroup
	rds:CreateDBSnapshot
	RDS: creato DB SubnetGroup

Prefisso del servizio	Azioni
	rds: Abbonamento CreateEvent
	rds: Cluster CreateGlobal
	rds: Gruppo CreateOption
	rds: DeleteBlue GreenDeployment
	RDS: DeletedB Backup ClusterAutomated
	RDS: gruppo DeleteDB ClusterParameter
	RDS: eliminato DB ClusterSnapshot
	RDS: DeletedB Backup InstanceAutomated
	RDS: elimina DB ParameterGroup
	rds:DeleteDBProxy
	RDS: elimina DB ProxyEndpoint
	RDS: elimina DB SecurityGroup
	rds:DeleteDBSnapshot
	RDS: elimina DB SubnetGroup
	rds: Abbonamento DeleteEvent
	rds: Cluster DeleteGlobal
	rds: Gruppo DeleteOption
	RDS: annulla la registrazione di DB ProxyTargets
	rds: Attributi DescribeAccount
	rds: DescribeBlue GreenDeployments
	rds: DescribeCertificates

Prefisso del servizio	Azioni
	ClusterAutomatedRDS: descrizione dei backup DB
	RDS: descritto B ClusterBacktracks
	RDS: descritto B ClusterEndpoints
	RDS: gruppi DB descritti ClusterParameter
	RDS: Descritto B ClusterParameters
	rds:DescribeDBClusters
	RDS:Attributi DescribeDB ClusterSnapshot
	RDS: Descrive DB ClusterSnapshots
	RDS: descritto B EngineVersions
	RDS: descrizione InstanceAutomated dei backup DB
	rds:DescribeDBInstances
	RDS: descritto B LogFiles
	RDS: descritto B ParameterGroups
	rds:DescribeDBParameters
	rds:DescribeDBProxies
	RDS: descritto B ProxyEndpoints
	RDS: gruppi DB descritti ProxyTarget
	RDS: Descritto B ProxyTargets
	RDS: raccomandazioni DB descritte
	RDS: descritto B SecurityGroups
	RDS: descritto B SnapshotAttributes

Prefisso del servizio	Azioni
	rds:DescribeDBSnapshots
	rds: Database DescribeDb SnapshotTenant
	rds: Descritto B SubnetGroups
	rds: Parametri DescribeEngine DefaultCluster
	rds: DescribeEngine DefaultParameters
	rds: Categorie DescribeEvent
	rds: DescribeEvents
	rds: abbonamenti DescribeEvent
	rds: Attività DescribeExport
	rds: Cluster DescribeGlobal
	rds: DescribeIntegrations
	rds: DescribeOption GroupOptions
	rds: Gruppi DescribeOption
	rds: DB DescribeOrderable InstanceOptions
	rds: DescribePending MaintenanceActions
	rds: dbInstances DescribeReserved
	rds: DB DescribeReserved InstancesOfferings
	rds: Regioni DescribeSource
	rds: Database DescribeTenant
	rds: DB DescribeValid InstanceModifications
	rds: DB DownloadComplete LogFile

Prefisso del servizio	Azioni
	LogFileRDS: scarica la parte DB
	rds:FailoverDBCluster
	rds: Cluster FailoverGlobal
	rds: Stream ModifyActivity
	rds: ModifyCertificates
	rds: DB ModifyCurrent ClusterCapacity
	RDS: modifica DB ClusterEndpoint
	RDS: gruppo ModifyDB ClusterParameter
	Attributo RDS:ModifyDB ClusterSnapshot
	RDS: ModifyDB ParameterGroup
	rds:ModifyDBProxy
	RDS: ModifyDB ProxyEndpoint
	RDS: gruppo ModifyDB ProxyTarget
	RDS: raccomandazione ModifyDB
	rds:ModifyDBSnapshot
	RDS: ModifyDB SnapshotAttribute
	RDS: ModifyDB SubnetGroup
	rds: Abbonamento ModifyEvent
	rds: Cluster ModifyGlobal
	rds: Gruppo ModifyOption
	rds: banca dati ModifyTenant

Prefisso del servizio	Azioni
	rds: DB PurchaseReserved InstancesOffering
	rds:RebootDBCluster
	RDS: registra DB ProxyTargets
	rds: RemoveFrom GlobalCluster
	rds: fromDBCluster RemoveRole
	rds: fromDBInstance RemoveRole
	rds: abbonamento RemoveSource IdentifierFrom
	ClusterParameterRDS: gruppo ResetDB
	RDS: resetDB ParameterGroup
	RDS: ripristina DB ClusterFrom S3
	RDS: ripristina l'istantanea DB ClusterFrom
	RDS: RestoreDB Time ClusterTo PointIn
	RDS: ripristina l'istantanea DB DB InstanceFrom
	RDS: ripristina DB S3 InstanceFrom
	RDS: RestoreDB Time InstanceTo PointIn
	RDS: ingresso DB revocato SecurityGroup
	rds: Stream StartActivity
	rds:StartDBCluster
	rds:StartDBInstance
	RDS: avvia DB InstanceAutomated BackupsReplication
	rds: Attività StartExport

Prefisso del servizio	Azioni
	rds: Stream StopActivity rds:StopDBCluster rds:StopDBInstance RDS: stopDB InstanceAutomated BackupsReplication rds: SwitchoverBlue GreenDeployment rds: Cluster SwitchoverGlobal rds: Replica SwitchoverRead

Prefisso del servizio	Azioni
redshift	spostamento verso il rosso: AcceptReserved NodeExchange spostamento verso il rosso: AddPartner spostamento verso il rosso: AssociateData ShareConsumer redshift: ingresso AuthorizeCluster SecurityGroup redshift: Condividi AuthorizeData redshift: Accesso AuthorizeEndpoint redshift: Accesso AuthorizeSnapshot redshift: BatchDelete ClusterSnapshots spostamento verso il rosso: BatchModify ClusterSnapshots spostamento verso il rosso: CancelResize redshift: istantanea CopyCluster redshift: Profilo CreateAuthentication redshift: CreateCluster spostamento verso il rosso: CreateCluster ParameterGroup spostamento verso il rosso: CreateCluster SecurityGroup redshift: istantanea CreateCluster spostamento verso il rosso: CreateCluster SubnetGroup spostamento verso il rosso: CreateCustom DomainAssociation redshift: Accesso CreateEndpoint redshift: abbonamento CreateEvent redshift: CreateHsm ClientCertificate

Prefisso del servizio	Azioni
	redshift: Configurazione CreateHsm redshift: CreateRedshift IdcApplication redshift: Azione CreateScheduled redshift: CreateSnapshot CopyGrant redshift: Pianifica CreateSnapshot redshift: Limite CreateUsage redshift: Condividi DeauthorizeData redshift: Profilo DeleteAuthentication redshift: DeleteCluster spostamento verso il rosso: DeleteCluster ParameterGroup spostamento verso il rosso: DeleteCluster SecurityGroup redshift: istantanea DeleteCluster spostamento verso il rosso: DeleteCluster SubnetGroup spostamento verso il rosso: DeleteCustom DomainAssociation redshift: Accesso DeleteEndpoint redshift: abbonamento DeleteEvent redshift: DeleteHsm ClientCertificate redshift: Configurazione DeleteHsm redshift: DeletePartner redshift: Azione DeleteScheduled redshift: DeleteSnapshot CopyGrant

Prefisso del servizio	Azioni
	redshift: Pianifica DeleteSnapshot redshift: Limite DeleteUsage redshift: Attributi DescribeAccount redshift: Profili DescribeAuthentication redshift: DescribeCluster DbRevisions spostamento verso il rosso: DescribeCluster ParameterGroups redshift: parametri DescribeCluster redshift: DescribeClusters spostamento verso il rosso: DescribeCluster SecurityGroups redshift: istantanee DescribeCluster redshift: DescribeCluster SubnetGroups redshift: Tracce DescribeCluster redshift: versioni DescribeCluster redshift: DescribeCustom DomainAssociations redshift: azioni DescribeData redshift: Consumatore DescribeData SharesFor redshift: Produttore DescribeData SharesFor redshift: DescribeDefault ClusterParameters redshift: Accesso DescribeEndpoint redshift: Autorizzazione DescribeEndpoint redshift: Categorie DescribeEvent

Prefisso del servizio	Azioni
	redshift: DescribeEvents redshift: abbonamenti DescribeEvent redshift: DescribeHsm ClientCertificates redshift: configurazioni DescribeHsm redshift: integrazioni DescribeInbound redshift: Stato DescribeLogging spostamento verso il rosso: DescribeNode ConfigurationOptions spostamento verso il rosso: DescribeOrderable ClusterOptions spostamento verso il rosso: DescribePartners spostamento verso il rosso: DescribeRedshift IdcApplications redshift: stato DescribeReserved NodeExchange spostamento verso il rosso: DescribeReserved NodeOfferings redshift: nodi DescribeReserved redshift: DescribeResize redshift: Azioni DescribeScheduled redshift: DescribeSnapshot CopyGrants redshift: Pianificazioni DescribeSnapshot spostamento verso il rosso: DescribeStorage spostamento verso il rosso: DescribeTable RestoreStatus redshift: Limiti DescribeUsage redshift: DisableLogging

Prefisso del servizio	Azioni
	redshift: copia DisableSnapshot redshift: DisassociateData ShareConsumer spostamento verso il rosso: EnableLogging redshift: copia EnableSnapshot redshift: Calcola FailoverPrimary redshift: credenziali GetCluster redshift: IAM GetCluster CredentialsWith spostamento verso il rosso: GetReserved NodeExchange Configura tionOptions redshift: Offerte GetReserved NodeExchange redshift: ListRecommendations redshift: Configurazione ModifyAqua redshift: Profilo ModifyAuthentication redshift: ModifyCluster spostamento verso il rosso: ModifyCluster DbRevision spostamento verso il rosso: ModifyCluster IamRoles redshift: Manutenzione ModifyCluster redshift: ModifyCluster ParameterGroup redshift: istantanea ModifyCluster spostamento verso il rosso: ModifyCluster SnapshotSchedule spostamento verso il rosso: ModifyCluster SubnetGroup

Prefisso del servizio	Azioni
	spostamento verso il rosso: ModifyCustom DomainAssociation redshift: Accesso ModifyEndpoint redshift: abbonamento ModifyEvent redshift: Azione ModifyScheduled redshift: Periodo ModifySnapshot CopyRetention redshift: Pianificazione ModifySnapshot redshift: Limite ModifyUsage spostamento verso il rosso: PauseCluster spostamento verso il rosso: PurchaseReserved NodeOffering spostamento verso il rosso: RebootCluster redshift: Condividi RejectData redshift: ResetCluster ParameterGroup spostamento verso il rosso: ResizeCluster spostamento verso il rosso: RestoreFrom ClusterSnapshot redshift: istantanea RestoreTable FromCluster spostamento verso il rosso: ResumeCluster redshift: ingresso RevokeCluster SecurityGroup redshift: Accesso RevokeEndpoint redshift: Accesso RevokeSnapshot redshift: Chiave RotateEncryption redshift: Stato UpdatePartner

Prefisso del servizio	Azioni
redshift-data	redshift-data: Dichiarazione BatchExecute
	redshift-data: CancelStatement
	dati redshift: DescribeStatement
	dati redshift: DescribeTable
	dati redshift: ExecuteStatement
	redshift-data: Risultato GetStatement
	redshift-data: ListDatabases
	dati redshift: ListSchemas
	dati redshift: ListStatements
	dati redshift: ListTables

Prefisso del servizio	Azioni
refactor-spaces	spazi refactorici: CreateApplication spazi di refattore: CreateEnvironment spazi di refattore: CreateRoute spazi di refattore: CreateService spazi di refattore: DeleteApplication spazi di refattore: DeleteEnvironment refactor-spaces: politica DeleteResource refactor-spaces: DeleteRoute spazi di refattore: DeleteService spazi di refattore: GetApplication spazi di refattore: GetEnvironment refactor-spaces: politica GetResource refactor-spaces: GetRoute spazi di refattore: GetService spazi di refattore: ListApplications spazi di refattore: ListEnvironments refactor-spaces: Vpc ListEnvironment spazi di refattore: ListRoutes spazi di refattore: ListServices refactor-spaces: politica PutResource refactor-spaces: UpdateRoute

Prefisso del servizio	Azioni
rekognition	riconoscimento: AssociateFaces
	riconoscimento: CompareFaces
	riconoscimento: versione CopyProject
	riconoscimento: CreateCollection
	riconoscimento: CreateDataset
	riconoscimento: CreateFace LivenessSession
	riconoscimento: CreateProject
	riconoscimento: versione CreateProject
	rekognition: Processore CreateStream
	riconoscimento: CreateUser
	riconoscimento: DeleteCollection
	riconoscimento: DeleteDataset
	riconoscimento: DeleteFaces
	riconoscimento: DeleteProject
	riconoscimento: politica DeleteProject
	riconoscimento: versione DeleteProject
	rekognition: Processore DeleteStream
	riconoscimento: DeleteUser
	riconoscimento: DescribeCollection
	riconoscimento: DescribeDataset
	riconoscimento: DescribeProjects

Prefisso del servizio	Azioni
	riconoscimento: versioni DescribeProject
	rekognition: Processore DescribeStream
	rekognition: etichette DetectCustom
	riconoscimento: DetectFaces
	riconoscimento: DetectLabels
	rekognition: etichette DetectModeration
	rekognition: Attrezzatura DetectProtective
	riconoscimento: DetectText
	riconoscimento: DisassociateFaces
	riconoscimento: iscrizioni DistributeDataset
	rekognition: Informazioni GetCelebrity
	rekognition: riconoscimento GetCelebrity
	riconoscimento: moderazione GetContent
	riconoscimento: GetFace rilevamento
	rekognition: risultati GetFace LivenessSession
	rekognition: Cerca GetFace
	rekognition: rilevamento GetLabel
	riconoscimento: GetMedia AnalysisJob
	riconoscimento: tracciamento GetPerson
	riconoscimento: rilevamento GetSegment
	riconoscimento: rilevamento GetText

Prefisso del servizio	Azioni
	riconoscimento: IndexFaces
	riconoscimento: ListCollections
	riconoscimento: iscrizioni ListDataset
	rekognition: Etichette ListDataset
	riconoscimento: ListFaces
	riconoscimento: ListMedia AnalysisJobs
	riconoscimento: politiche ListProject
	riconoscimento: processori ListStream
	riconoscimento: ListUsers
	riconoscimento: politica PutProject
	riconoscimento: RecognizeCelebrities
	riconoscimento: SearchFaces
	riconoscimento: SearchFaces ByImage
	riconoscimento: SearchUsers
	riconoscimento: SearchUsers ByImage
	riconoscimento: riconoscimento StartCelebrity
	riconoscimento: moderazione StartContent
	riconoscimento: StartFace rilevamento
	riconoscimento: StartFace LivenessSession
	rekognition: Cerca StartFace
	rekognition: rilevamento StartLabel

Prefisso del servizio	Azioni
	riconoscimento: StartMedia AnalysisJob
	riconoscimento: tracciamento StartPerson
	rekognition: versione StartProject
	rekognition: rilevamento StartSegment
	rekognition: Processore StartStream
	rekognition: rilevamento StartText
	riconoscimento: versione StopProject
	rekognition: Processore StopStream
	rekognition: iscrizioni UpdateDataset
	rekognition: Processore UpdateStream

Prefisso del servizio	Azioni
resiliencehub	hub di resilienza: AddDraft AppVersion ResourceMappings hub di resilienza: CreateApp resiliencehub: Componente CreateApp VersionApp resiliencehub: CreateApp VersionResource resiliencehub: modello CreateRecommendation resiliencehub: politica CreateResiliency hub di resilienza: DeleteApp resiliencehub: valutazione DeleteApp resiliencehub: DeleteApp InputSource resiliencehub: Componente DeleteApp VersionApp resiliencehub: DeleteApp VersionResource resiliencehub: modello DeleteRecommendation resiliencehub: politica DeleteResiliency hub di resilienza: DescribeApp resiliencehub: valutazione DescribeApp resiliencehub: versione DescribeApp resiliencehub: Componente DescribeApp VersionApp resiliencehub: DescribeApp VersionResource hub di resilienza: DescribeApp VersionResources ResolutionStatus hub di resilienza: DescribeApp VersionTemplate resiliencehub: Stato DescribeDraft AppVersion ResourcesImport

Prefisso del servizio	Azioni
	resiliencehub: Politica DescribeResiliency hub di resilienza: ImportResources ToDraft AppVersion resiliencehub: raccomandazioni ListAlarm resiliencehub: valutazioni ListApp hub di resilienza: ListApp ComponentCompliances hub di resilienza: ListApp ComponentRecommendations hub di resilienza: ListApp InputSources hub di resilienza: ListApps resiliencehub: Componenti ListApp VersionApp resiliencehub: mappature ListApp VersionResource hub di resilienza: ListApp VersionResources resiliencehub: versioni ListApp resiliencehub: modelli ListRecommendation resiliencehub: politiche ListResiliency resiliencehub: raccomandazioni ListSop resiliencehub: ListSuggested ResiliencyPolicies resiliencehub: raccomandazioni ListTest resiliencehub: Risorse ListUnsupported AppVersion resiliencehub: versione PublishApp resiliencehub: Modello PutDraft AppVersion hub di resilienza: RemoveDraft AppVersion ResourceMappings

Prefisso del servizio	Azioni
	hub di resilienza: ResolveApp VersionResources resiliencehub: valutazione StartApp resiliencehub: UpdateApp resiliencehub: versione UpdateApp resiliencehub: Componente UpdateApp VersionApp resiliencehub: UpdateApp VersionResource resiliencehub: politica UpdateResiliency

Prefisso del servizio	Azioni
resource-explorer-2	resource-explorer-2: Visualizza AssociateDefault
	resource-explorer-2: Visualizza BatchGet
	resource-explorer-2: CreateIndex
	esploratore-risorsa-2: CreateView
	esploratore-risorsa-2: DeleteIndex
	esploratore-risorsa-2: DeleteView
	resource-explorer-2: Visualizza DisassociateDefault
	resource-explorer-2: Configurazione GetAccount LevelService
	resource-explorer-2: Visualizza GetDefault
	resource-explorer-2: GetIndex
	esploratore-risorsa-2: ListIndexes
	esploratore-risorsa-2: ListIndexes ForMembers
	esploratore-risorsa-2: ListSupported ResourceTypes
	esploratore-risorsa-2: ListViews
	resource-explorer-2:Search
	resource-explorer-2: Tipo UpdateIndex
	resource-explorer-2: UpdateView

Prefisso del servizio	Azioni
resource-groups	gruppi di risorse: CreateGroup gruppi di risorse: DeleteGroup resource-groups: Impostazioni GetAccount gruppi di risorse: GetGroup resource-groups: configurazione GetGroup resource-groups: GetGroup Interrogazione gruppi di risorse: GroupResources gruppi di risorse: ListGroup Risorse gruppi di risorse: ListGroups resource-groups: configurazione PutGroup gruppi-risorse: SearchResources gruppi di risorse: UngroupResources resource-groups: Impostazioni UpdateAccount gruppi di risorse: UpdateGroup gruppi di risorse: Interrogazione UpdateGroup

Prefisso del servizio	Azioni
robomaker	robomaker: BatchDelete Mondi
	robomaker: BatchDescribe SimulationJob
	robomaker: Job CancelDeployment
	robomaker: Job CancelSimulation
	robomaker: CancelSimulation JobBatch
	robomaker: CancelWorld ExportJob
	robomaker: CancelWorld GenerationJob
	robomaker: Job CreateDeployment
	robomaker: CreateFleet
	robomaker: CreateRobot
	robomaker: Applicazione CreateRobot
	robomaker: CreateRobot ApplicationVersion
	robomaker: Applicazione CreateSimulation
	robomaker: CreateSimulation ApplicationVersion
	robomaker: Job CreateSimulation
	robomaker: CreateWorld ExportJob
	robomaker: CreateWorld GenerationJob
	robomaker: Modello CreateWorld
	robomaker: DeleteFleet
	robomaker: DeleteRobot
	robomaker: Applicazione DeleteRobot

Prefisso del servizio	Azioni
	robomaker: Applicazione DeleteSimulation
	robomaker: Modello DeleteWorld
	robomaker: DeregisterRobot
	robomaker: Job DescribeDeployment
	robomaker: DescribeFleet
	robomaker: DescribeRobot
	robomaker: Applicazione DescribeRobot
	robomaker: Applicazione DescribeSimulation
	robomaker: Job DescribeSimulation
	robomaker: DescribeSimulation JobBatch
	robomaker: DescribeWorld
	robomaker: DescribeWorld ExportJob
	robomaker: DescribeWorld GenerationJob
	robomaker: Modello DescribeWorld
	robomaker: GetWorld TemplateBody
	robomaker: Offerte di lavoro ListDeployment
	robomaker: ListFleets
	robomaker: Applicazioni ListRobot
	robomaker: ListRobots
	robomaker: Applicazioni ListSimulation
	robomaker: ListSimulation JobBatches

Prefisso del servizio	Azioni
	robomaker: Offerte di lavoro ListSimulation
	robomaker: ListWorld ExportJobs
	robomaker: ListWorld GenerationJobs
	robomaker: ListWorlds
	robomaker: modelli ListWorld
	robomaker: RegisterRobot
	robomaker: Job RestartSimulation
	robomaker: StartSimulation JobBatch
	robomaker: Job SyncDeployment
	robomaker: Applicazione UpdateRobot
	robomaker: Applicazione UpdateSimulation
	robomaker: Modello UpdateWorld

Prefisso del servizio	Azioni
rolesanywhere	ruoli ovunque: CreateProfile rolesanywhere: Anchor CreateTrust rolesanywhere: mappatura DeleteAttribute ruoli ovunque: DeleteCrl ruoli ovunque: DeleteProfile rolesanywhere: Anchor DeleteTrust ruoli ovunque: DisableCrl ruoli ovunque: DisableProfile rolesanywhere: Anchor DisableTrust ruoli ovunque: EnableCrl ruoli ovunque: EnableProfile rolesanywhere: Anchor EnableTrust ruoli ovunque: GetCrl ruoli ovunque: GetProfile ruoli ovunque: GetSubject rolesanywhere: Anchor GetTrust ruoli ovunque: ImportCrl ruoli ovunque: ListCrls ruoli ovunque: ListProfiles ruoli ovunque: ListSubjects rolesanywhere: Anchors ListTrust

Prefisso del servizio	Azioni
	rolesanywhere: mappatura PutAttribute rolesanywhere: Impostazioni PutNotification rolesanywhere: Impostazioni ResetNotification ruoli ovunque: UpdateCrl ruoli ovunque: UpdateProfile rolesanywhere: Anchor UpdateTrust

Prefisso del servizio	Azioni
route53	percorso 53: ActivateKey SigningKey Route 53: associa una zona VPC WithHosted route53: Collezione ChangeCidr route 53: ChangeResource RecordSets route53: Collezione CreateCidr route53: Verifica CreateHealth route53: Zona CreateHosted percorso 53: CreateKey SigningKey percorso 53: CreateQuery LoggingConfig percorso 53: CreateReusable DelegationSet route53: politica CreateTraffic percorso 53: CreateTraffic PolicyInstance percorso 53: CreateTraffic PolicyVersion Route 53: crea VPC AssociationAuthorization percorso 53: DeactivateKey SigningKey route53: Collezione DeleteCidr route53: Verifica DeleteHealth route53: Zona DeleteHosted percorso 53: DeleteKey SigningKey percorso 53: DeleteQuery LoggingConfig percorso 53: DeleteReusable DelegationSet

Prefisso del servizio	Azioni
	route53: politica DeleteTraffic
	percorso 53: DeleteTraffic PolicyInstance
	percorso 53: elimina VPC AssociationAuthorization
	route 53: ZoneDNSsec DisableHosted
	Route 53FromHosted: dissocia VPC Zone
	route 53: Zone DNSsec EnableHosted
	route53GetAccount: Limite
	percorso 53: GetChange
	percorso 53: GetChecker IpRanges
	route53:GetDNSSEC
	route53: Posizione GetGeo
	route53: Verifica GetHealth
	percorso 53: GetHealth CheckCount
	percorso 53: GetHealth CheckLast FailureReason
	percorso 53: GetHealth CheckStatus
	route53: Zona GetHosted
	percorso 53: GetHosted ZoneCount
	percorso 53: GetHosted ZoneLimit
	percorso 53: GetQuery LoggingConfig
	percorso 53: GetReusable DelegationSet
	route53: Limite GetReusable DelegationSet

Prefisso del servizio	Azioni
	route53: Politica GetTraffic
	percorso 53: GetTraffic PolicyInstance
	route53: conta GetTraffic PolicyInstance
	route53: Blocchi ListCidr
	route53: Collezioni ListCidr
	route53: sedi ListCidr
	route53: Sedi ListGeo
	route53: controlli ListHealth
	route53: Zone ListHosted
	route53: Nome ListHosted ZonesBy
	route 53: VPC ListHosted ZonesBy
	percorso 53: ListQuery LoggingConfigs
	percorso 53: ListResource RecordSets
	percorso 53: ListReusable DelegationSets
	route53: politiche ListTraffic
	percorso 53: ListTraffic PolicyInstances
	route53: Zona ListTraffic PolicyInstances ByHosted
	percorso 53: ListTraffic PolicyInstances ByPolicy
	percorso 53: ListTraffic PolicyVersions
	percorso 53: elenco VPC AssociationAuthorizations
	route53:TestDNSAnswer

Prefisso del servizio	Azioni
	route53: Verifica UpdateHealth percorso 53: UpdateHosted ZoneComment percorso 53: UpdateTraffic PolicyComment percorso 53: UpdateTraffic PolicyInstance

Prefisso del servizio	Azioni
route53-recovery-control-config	route53-recovery-control-config: CreateCluster route53-recovery-control-config: Pannello CreateControl route53-recovery-control-config: Controllo CreateRouting route53-recovery-control-config: Regola CreateSafety route53-recovery-control-config: DeleteCluster route53-recovery-control-config: Pannello DeleteControl route53-recovery-control-config: Controllo DeleteRouting route53-recovery-control-config: Regola DeleteSafety route53-recovery-control-config: DescribeCluster route53-recovery-control-config: Pannello DescribeControl route53-recovery-control-config: Controllo DescribeRouting route53-recovery-control-config: Regola DescribeSafety route53-recovery-control-config: politica GetResource route53-recovery-control-config: Route 53 ListAssociated HealthChecks route53-recovery-control-config: ListClusters route53-recovery-control-config: Pannelli ListControl route53-recovery-control-config: Controlli ListRouting route53-recovery-control-config: Regole ListSafety route53-recovery-control-config: Pannello UpdateControl route53-recovery-control-config: Controllo UpdateRouting

Prefisso del servizio	Azioni
	route53-recovery-control-config: Regola UpdateSafety

Prefisso del servizio	Azioni
route53-recovery-readiness	route53-recovery-ready: CreateCell predisposizione al ripristino del route53: CreateCross AccountAuthorization route53-recovery-readiness: verifica CreateReadiness route53-recovery-readiness: Gruppo CreateRecovery route53-recovery-readiness: impostato CreateResource route53-prontezza per il ripristino: DeleteCell predisposizione al ripristino del route53: DeleteCross AccountAuthorization route53-recovery-readiness: verifica DeleteReadiness route53-recovery-readiness: Gruppo DeleteRecovery route53-recovery-readiness: impostato DeleteResource route53-recovery-readiness: raccomandazioni GetArchitecture route53 - predisposizione al ripristino: GetCell predisposizione al ripristino del route53: GetCell Readiness Summary route53-recovery-readiness: verifica GetReadiness route53-recovery-readiness: Stato GetReadiness CheckResource predisposizione al ripristino del route53: GetReadiness CheckStatus route53-recovery-readiness: Gruppo GetRecovery route53-recovery-readiness: riepilogo GetRecovery GroupReadiness

Prefisso del servizio	Azioni
	route53-recovery-readiness: impostato GetResource
	route53-prontezza per il ripristino: ListCells
	predisposizione al ripristino del route53: ListCross AccountAu thorizations
	route53-recovery-readiness: verifiche ListReadiness
	route53-recovery-readiness: gruppi ListRecovery
	route53-recovery-readiness: set ListResource
	route53-prontezza per il ripristino: ListRules
	predisposizione al ripristino del route53: UpdateCell
	route53-recovery-readiness: verifica UpdateReadiness
	route53-recovery-readiness: Gruppo UpdateRecovery
	route53-recovery-readiness: impostato UpdateResource

Prefisso del servizio	Azioni
route53resolver	resolver route53: AssociateFirewall RuleGroup route53resolver: Indirizzo AssociateResolver EndpointIp route53resolver: Config AssociateResolver QueryLog route53resolver: AssociateResolver Regola route53resolver: CreateFirewall DomainList route53resolver: regola CreateFirewall route53resolver: CreateFirewall RuleGroup route53resolver: punto terminale CreateResolver route53resolver: Config CreateResolver QueryLog route53resolver: CreateResolver Regola route53resolver: DeleteFirewall DomainList route53resolver: regola DeleteFirewall route53resolver: DeleteFirewall RuleGroup route53resolver: Risolutore DeleteOutpost route53resolver: endpoint DeleteResolver route53resolver: Config DeleteResolver QueryLog route53resolver: DeleteResolver Regola route53resolver: DisassociateFirewall RuleGroup route53resolver: Indirizzo DisassociateResolver EndpointIp route53resolver: Config DisassociateResolver QueryLog route53resolver: DisassociateResolver Regola

Prefisso del servizio	Azioni
	route53resolver: Config GetFirewall resolver route53: GetFirewall DomainList resolver route53: GetFirewall RuleGroup route53resolver: Associazione GetFirewall RuleGroup route53resolver: politica GetFirewall RuleGroup route53resolver: GetOutpost Risolutore route53resolver: Config GetResolver resolver route53: GetResolver DnssecConfig route53resolver: punto terminale GetResolver route53resolver: Config GetResolver QueryLog resolver route53: GetResolver QueryLog ConfigAssociation resolver route53: GetResolver QueryLog ConfigPolicy route53resolver: regola GetResolver route53resolver: GetResolver RuleAssociation resolver route53: GetResolver RulePolicy route53resolver: Domini ImportFirewall route53resolver: ListFirewall configurazioni route53resolver: ListFirewall DomainLists route53resolver: Domini ListFirewall route53resolver: ListFirewall RuleGroup Associazioni route53 resolver: ListFirewall RuleGroups

Prefisso del servizio	Azioni
	route53resolver: regole ListFirewall route53resolver: ListOutpost risolutori route53resolver: ListResolver configurazioni route53resolver: ListResolver DnssecConfigs route53resolver: indirizzi ListResolver EndpointIp route53resolver: endpoint ListResolver resolver route53: ListResolver QueryLog ConfigAssociations route53resolver: configurazioni ListResolver QueryLog route53resolver: ListResolver RuleAssociations route53resolver: regole ListResolver route53resolver: politica PutFirewall RuleGroup resolver route53: PutResolver QueryLog ConfigPolicy route53resolver: Config UpdateFirewall route53resolver: Domini UpdateFirewall route53resolver: UpdateFirewall Regola route53resolver: Associazione UpdateFirewall RuleGroup route53resolver: UpdateOutpost Risolutore route53resolver: Config UpdateResolver resolver route53: UpdateResolver DnssecConfig route53resolver: punto terminale UpdateResolver route53resolver: UpdateResolver Regola

Prefisso del servizio	Azioni
rum	rum: BatchCreate RumMetric definizioni rum: BatchDelete RumMetric definizioni rum: BatchGet RumMetric definizioni rum: CreateApp Monitor rum: DeleteApp monitor tamburo: DeleteRum MetricsDestination rum: GetApp monitor tamburo: GetApp MonitorData rum: ListApp monitor tamburo: ListRum MetricsDestinations rum: PutRum MetricsDestination rum: UpdateApp monitor tamburo: UpdateRum MetricDefinition

Prefisso del servizio	Azioni
s3	s3: Centro AssociateAccess GrantsIdentity s3: Sovvenzione CreateAccess s3: CreateAccess GrantsInstance s3: CreateAccess GrantsLocation s3: Punto CreateAccess s3: CreateAccess PointFor ObjectLambda s3: CreateBucket s3: CreateJob s3: Punto CreateMulti RegionAccess s3: Concessione DeleteAccess s3: DeleteAccess GrantsInstance s3: DeleteAccess GrantsInstance ResourcePolicy s3: DeleteAccess GrantsLocation s3: Punto DeleteAccess s3: DeleteAccess PointFor ObjectLambda s3: DeleteAccess PointPolicy s3: DeleteAccess PointPolicy ForObject Lambda s3: Blocca PutAccount PublicAccess s3: DeleteBucket s3: Configurazione PutAnalytics s3: CORS PutBucket

Prefisso del servizio	Azioni
	s3: Configurazione PutEncryption s3: PutIntelligent TieringConfiguration s3: Configurazione PutInventory s3: Configurazione PutLifecycle s3: Configurazione PutMetrics s3: PutBucket OwnershipControls s3: Politica DeleteBucket s3: Blocca PutBucket PublicAccess s3: Configurazione PutReplication s3: Sito Web DeleteBucket s3: Punto DeleteMulti RegionAccess s3: DeleteStorage LensConfiguration s3: DescribeJob s3: DescribeMulti RegionAccess PointOperation s3: Centro DissociateAccess GrantsIdentity s3: Configurazione GetAccelerate s3: Concessione GetAccess s3: GetAccess GrantsInstance s3: GetAccess GrantsInstance ForPrefix s3: GetAccess GrantsInstance ResourcePolicy s3: GetAccess GrantsLocation

Prefisso del servizio	Azioni
	s3: Punto GetAccess s3: GetAccess PointConfiguration ForObject Lambda s3: GetAccess PointFor ObjectLambda s3: GetAccess PointPolicy s3: GetAccess PointPolicy ForObject Lambda s3: Stato GetAccess PointPolicy s3: GetAccess PointPolicy StatusFor ObjectLambda s3: Blocca GetAccount PublicAccess s3: Acl GetBucket s3: Configurazione GetAnalytics s3: CORS GetBucket s3: Configurazione GetEncryption s3: GetIntelligent TieringConfiguration s3: Configurazione GetInventory s3: Configurazione GetLifecycle s3: Ubicazione GetBucket s3: Registrazione GetBucket s3: Configurazione GetMetrics s3: Notifica GetBucket s3: Configurazione GetBucket ObjectLock s3: GetBucket OwnershipControls

Prefisso del servizio	Azioni
	s3: Politica GetBucket s3: GetBucket PolicyStatus s3: Blocca GetBucket PublicAccess s3: Configurazione GetReplication s3: GetBucket RequestPayment s3: controllo delle versioni GetBucket s3: Sito Web GetBucket s3: Accesso GetData s3: Punto GetMulti RegionAccess s3: GetMulti RegionAccess PointPolicy s3: Stato GetMulti RegionAccess PointPolicy s3: GetMulti RegionAccess PointRoutes s3: attributi GetObject s3: GetStorage LensConfiguration s3: GetStorage LensDashboard s3: Sovvenzioni ListAccess s3: ListAccess GrantsInstances s3: ListAccess GrantsLocations s3: Punti ListAccess s3: ListAccess PointsFor ObjectLambda s3: ListAll MyBuckets

Prefisso del servizio	Azioni
	s3: ListJobs s3: ListBucket MultipartUploads s3: Punti ListMulti RegionAccess s3: ListStorage LensConfigurations s3: Configurazione PutAccelerate s3: PutAccess GrantsInstance ResourcePolicy s3: PutAccess PointConfiguration ForObject Lambda s3: PutAccess PointPolicy s3: PutAccess PointPolicy ForObject Lambda s3: Blocca PutAccount PublicAccess s3: Acl PutBucket s3: Configurazione PutAnalytics s3: CORS PutBucket s3: Configurazione PutEncryption s3: PutIntelligent TieringConfiguration s3: Configurazione PutInventory s3: Configurazione PutLifecycle s3: Registrazione PutBucket s3: Configurazione PutMetrics s3: Notifica PutBucket s3: Configurazione PutBucket ObjectLock

Prefisso del servizio	Azioni
	s3: PutBucket OwnershipControls s3: Politica PutBucket s3: Blocca PutBucket PublicAccess s3: Configurazione PutReplication s3: PutBucket RequestPayment s3: controllo delle versioni PutBucket s3: Sito Web PutBucket s3: PutMulti RegionAccess PointPolicy s3: PutStorage LensConfiguration s3: SubmitMulti RegionAccess PointRoutes s3: UpdateAccess GrantsLocation s3: Priorità UpdateJob s3: Stato UpdateJob
s3-outposts	avamposti s3: CreateEndpoint avamposti s3: DeleteEndpoint avamposti s3: ListEndpoints s3-outposts: con S3 ListOutposts s3-outposts: ListShared punti finali

Prefisso del servizio	Azioni
sagemaker-geospatial	sagemaker-geospatial: DeleteEarth ObservationJob sagemaker geospaziale: DeleteVector EnrichmentJob sagemaker geospaziale: ExportEarth ObservationJob sagemaker geospaziale: ExportVector EnrichmentJob sagemaker geospaziale: GetEarth ObservationJob sagemaker geospaziale: GetRaster DataCollection sagemaker geospaziale: GetTile sagemaker geospaziale: GetVector EnrichmentJob sagemaker geospaziale: ListEarth ObservationJobs sagemaker geospaziale: ListRaster DataCollections sagemaker geospaziale: ListVector EnrichmentJobs sagemaker geospaziale: SearchRaster DataCollection sagemaker geospaziale: StartEarth ObservationJob sagemaker geospaziale: StartVector EnrichmentJob sagemaker geospaziale: StopEarth ObservationJob sagemaker geospaziale: StopVector EnrichmentJob

Prefisso del servizio	Azioni
savingsplans	piani di risparmio: Pianifica CreateSavings
	piani di risparmio: DeleteQueued SavingsPlan
	piani di risparmio: DescribeSavings PlanRates
	savingplans: piani DescribeSavings
	piani di risparmio: Tariffe DescribeSavings PlansOffering
	piani di risparmio: DescribeSavings PlansOfferings
	savingsplans: Pianifica ReturnSavings

Prefisso del servizio	Azioni
schemas	schemi: CreateDiscoverer schemi: CreateRegistry schemi: CreateSchema schemi: DeleteDiscoverer schemi: DeleteRegistry schemi: politica DeleteResource schemi: DeleteSchema schemi: versione DeleteSchema schemi: vincolante DescribeCode schemi: DescribeDiscoverer schemi: DescribeRegistry schemi: DescribeSchema schemi: ExportSchema schemi: GetCode BindingSource schemi: schema GetDiscovered schemi: politica GetResource schemi: ListDiscoverers schemi: ListRegistries schemi: ListSchemas schemi: versioni ListSchema schemi: vincolante PutCode

Prefisso del servizio	Azioni
	schemi: politica PutResource schemi: SearchSchemas schemi: StartDiscoverer schemi: StopDiscoverer schemi: UpdateDiscoverer schemi: UpdateRegistry schemi: UpdateSchema
sdb	sdb: CreateDomain sdb: DeleteDomain sdb: DomainMetadata sdb: ListDomains

Prefisso del servizio	Azioni
secretsmanager	gestore dei segreti: Segreto CancelRotate gestore dei segreti: CreateSecret secretsmanager: politica DeleteResource gestore dei segreti: DeleteSecret gestore dei segreti: DescribeSecret secretsmanager: password GetRandom secretsmanager: Politica GetResource secretsmanager: Valore GetSecret gestore dei segreti: ListSecrets gestore dei segreti: ListSecret VersionIds secretsmanager: Politica PutResource secretsmanager: Valore PutSecret gestore dei segreti: RemoveRegions FromReplication gestore dei segreti: ReplicateSecret ToRegions gestore dei segreti: RestoreSecret gestore dei segreti: RotateSecret gestore dei segreti: StopReplication ToReplica gestore dei segreti: UpdateSecret secretsmanager: politica ValidateResource

Prefisso del servizio	Azioni
securityhub	securityhub: invito AcceptAdministrator
	hub di sicurezza: AcceptInvitation
	hub di sicurezza: BatchDelete AutomationRules
	securityhub: standard BatchDisable
	securityhub: Standard BatchEnable
	hub di sicurezza: BatchGet AutomationRules
	securityhub: Associazioni BatchGet ConfigurationPolicy
	hub di sicurezza: BatchGet SecurityControls
	securityhub: Associazioni BatchGet StandardsControl
	securityhub: risultati BatchImport
	hub di sicurezza: BatchUpdate AutomationRules
	securityhub: risultati BatchUpdate
	securityhub: Associazioni BatchUpdate StandardsControl
	securityhub: Target CreateAction
	securityhub: Regola CreateAutomation
	securityhub: politica CreateConfiguration
	securityhub: Aggregatore CreateFinding
	hub di sicurezza: CreateInsight
	hub di sicurezza: CreateMembers
	hub di sicurezza: DeclineInvitations
	securityhub: Target DeleteAction

Prefisso del servizio	Azioni
	securityhub: politica DeleteConfiguration
	securityhub: Aggregatore DeleteFinding
	hub di sicurezza: DeleteInsight
	hub di sicurezza: DeleteInvitations
	hub di sicurezza: DeleteMembers
	securityhub: obiettivi DescribeAction
	hub di sicurezza: DescribeHub
	securityhub: Configurazione DescribeOrganization
	hub di sicurezza: DescribeProducts
	hub di sicurezza: DescribeStandards
	securityhub: Prodotto DisableImport FindingsFor
	hub di sicurezza: DisableOrganization AdminAccount
	securityhub: hub DisableSecurity
	hub di sicurezza: DisassociateFrom AdministratorAccount
	hub di sicurezza: DisassociateFrom MasterAccount
	hub di sicurezza: DisassociateMembers
	securityhub: Prodotto EnableImport FindingsFor
	hub di sicurezza: EnableOrganization AdminAccount
	securityhub: hub EnableSecurity
	securityhub: Conto GetAdministrator
	securityhub: politica GetConfiguration

Prefisso del servizio	Azioni
	hub di sicurezza: GetConfiguration PolicyAssociation
	securityhub: standard GetEnabled
	securityhub: Aggregatore GetFinding
	securityhub: Storia GetFinding
	hub di sicurezza: GetFindings
	securityhub: risultati GetInsight
	hub di sicurezza: GetInsights
	securityhub: conta GetInvitations
	securityhub: Conto GetMaster
	hub di sicurezza: GetMembers
	hub di sicurezza: GetSecurity ControlDefinition
	hub di sicurezza: InviteMembers
	securityhub: Regole ListAutomation
	securityhub: politiche ListConfiguration
	hub di sicurezza: ListConfiguration PolicyAssociations
	securityhub: Importa ListEnabled ProductsFor
	securityhub: aggregatori ListFinding
	hub di sicurezza: ListInvitations
	hub di sicurezza: ListMembers
	hub di sicurezza: ListOrganization AdminAccounts
	hub di sicurezza: ListSecurity ControlDefinitions

Prefisso del servizio	Azioni
	hub di sicurezza: ListStandards ControlAssociations
	hub di sicurezza: StartConfiguration PolicyAssociation
	hub di sicurezza: StartConfiguration PolicyDisassociation
	securityhub: Obiettivo UpdateAction
	securityhub: politica UpdateConfiguration
	securityhub: Aggregatore UpdateFinding
	hub di sicurezza: UpdateFindings
	hub di sicurezza: UpdateInsight
	securityhub: Configurazione UpdateOrganization
	securityhub: controllo UpdateSecurity
	hub di sicurezza: UpdateSecurity HubConfiguration

Prefisso del servizio	Azioni
securitylake	lago di sicurezza: CreateAws LogSource lago di sicurezza: CreateCustom LogSource securitylake: abbonamento CreateData LakeException securitylake: Configurazione CreateData LakeOrganization securitylake: CreateSubscriber securitylake: notifica CreateSubscriber securitylake: DeleteAws LogSource lago di sicurezza: DeleteCustom LogSource securitylake: abbonamento DeleteData LakeException securitylake: Configurazione DeleteData LakeOrganization securitylake: DeleteSubscriber securitylake: notifica DeleteSubscriber securitylake: Amministratore DeregisterData LakeDelegated securitylake: Abbonamento GetData LakeException securitylake: Configurazione GetData LakeOrganization securitylake: GetData LakeSources lago di sicurezza: GetSubscriber securitylake: laghi ListData securitylake: Fonti ListLog lago di sicurezza: ListSubscribers securitylake: amministratore RegisterData LakeDelegated

Prefisso del servizio	Azioni
	securitylake: Abbonamento UpdateData LakeException securitylake: UpdateSubscriber securitylake: notifica UpdateSubscriber
serverlessrepo	repository senza server: CreateApplication serverlessrepo: versione CreateApplication serverlessrepo: impostato CreateCloud FormationChange repository senza server: CreateCloud FormationTemplate repository senza server: DeleteApplication repository senza server: GetApplication serverlessrepo: politica GetApplication repository senza server: GetCloud FormationTemplate serverlessrepo: dipendenze ListApplication repository senza server: ListApplications serverlessrepo: versioni ListApplication serverlessrepo: politica PutApplication repository senza server: UnshareApplication repository senza server: UpdateApplication

Prefisso del servizio	Azioni
servicecatalog	servicecatalog: Condividi AcceptPortfolio catalogo dei servizi: AssociateBudget WithResource catalogo dei servizi: AssociatePrincipal WithPortfolio catalogo dei servizi: AssociateProduct WithPortfolio catalogo dei servizi: AssociateService ActionWith ProvisioningArtifact service catalog: Artifact BatchAssociate ServiceAction WithProvisioning service catalog: Artifact BatchDisassociate ServiceAction FromProvisioning catalogo dei servizi: CopyProduct catalogo dei servizi: CreateConstraint catalogo dei servizi: CreatePortfolio servicecatalog: Condividi CreatePortfolio catalogo dei servizi: CreateProduct catalogo dei servizi: CreateProvisioned ProductPlan service catalog: Artifact CreateProvisioning servicecatalog: CreateService Azione service catalog: DeleteConstraint catalogo dei servizi: DeletePortfolio servicecatalog: Condividi DeletePortfolio catalogo dei servizi: DeleteProduct

Prefisso del servizio	Azioni
	catalogo dei servizi: DeleteProvisioned ProductPlan service catalog: Artifact DeleteProvisioning servicecatalog: DeleteService Azione service catalog: DescribeConstraint catalogo dei servizi: DescribeCopy ProductStatus catalogo dei servizi: DescribePortfolio servicecatalog: condivisioni DescribePortfolio catalogo dei servizi: DescribePortfolio ShareStatus catalogo dei servizi: DescribeProduct catalogo dei servizi: DescribeProduct AsAdmin servicecatalog: Visualizza DescribeProduct catalogo dei servizi: DescribeProvisioned ProductPlan service catalog: Artifact DescribeProvisioning servicecatalog: DescribeProvisioning Parametri service catalog: DescribeRecord servicecatalog: DescribeService Azione servicecatalog: DescribeService ActionExecution Parametri Service Catalog: disabilita AWSOrganizationsAccess catalogo dei servizi: DisassociateBudget FromResource catalogo dei servizi: DisassociatePrincipal FromPortfolio catalogo dei servizi: DisassociateProduct FromPortfolio

Prefisso del servizio	Azioni
	catalogo dei servizi: DisassociateService ActionFrom Provision ingArtifact Catalogo dei servizi: abilita AWSOrganizationsAccess catalogo dei servizi: ExecuteProvisioned ProductPlan servicecatalog: ExecuteProvisioned ProductService Azione Catalogo dei servizi: GET AWSOrganizationsAccessStatus catalogo dei servizi: GetProvisioned ProductOutputs catalogo dei servizi: ImportAs ProvisionedProduct catalogo dei servizi: ListAccepted PortfolioShares catalogo dei servizi: ListBudgets ForResource catalogo dei servizi: ListConstraints ForPortfolio servicecatalog: Percorsi ListLaunch service catalog: ListOrganization PortfolioAccess servicecatalog: Accesso ListPortfolio catalogo dei servizi: ListPortfolios catalogo dei servizi: ListPortfolios ForProduct catalogo dei servizi: ListPrincipals ForPortfolio catalogo dei servizi: ListProvisioned ProductPlans servicecatalog: artefatti ListProvisioning catalogo dei servizi: ListProvisioning ArtifactsFor ServiceAction servicecatalog: Storia ListRecord

Prefisso del servizio	Azioni
	servicecatalog: ListService Azioni servicecatalog: ListService ActionsFor ProvisioningArtifact catalogo dei servizi: ListStack InstancesFor ProvisionedProduct catalogo dei servizi: NotifyProvision ProductEngine WorkflowResult servicecatalog: Risultato NotifyTerminate ProvisionedProduct EngineWorkflow servicecatalog: NotifyUpdate ProvisionedProduct EngineWorkflow Risultato service catalog: ProvisionProduct servicecatalog: Condividi RejectPortfolio servicecatalog: ScanProvisioned Prodotti catalogo dei servizi: SearchProducts catalogo dei servizi: SearchProducts AsAdmin servicecatalog: Prodotti SearchProvisioned servicecatalog: TerminateProvisioned Prodotto catalogo dei servizi: UpdateConstraint catalogo dei servizi: UpdatePortfolio servicecatalog: Condividi UpdatePortfolio catalogo dei servizi: UpdateProduct servicecatalog: Prodotto UpdateProvisioned catalogo dei servizi: UpdateProvisioned ProductProperties service catalog: Artifact UpdateProvisioning

Prefisso del servizio	Azioni
	servicecatalog: UpdateService Azione

Prefisso del servizio	Azioni
servicediscovery	servicediscovery: CreateHttp Namespace
	scoperta del servizio: CreatePrivate DnsNamespace
	individuazione del servizio: CreatePublic DnsNamespace
	individuazione del servizio: CreateService
	individuazione del servizio: DeleteNamespace
	individuazione del servizio: DeleteService
	individuazione del servizio: DeregisterInstance
	individuazione del servizio: GetInstance
	individuazione del servizio: GetInstances HealthStatus
	individuazione del servizio: GetNamespace
	individuazione del servizio: GetOperation
	individuazione del servizio: GetService
	individuazione del servizio: ListInstances
	individuazione del servizio: ListNamespaces
	individuazione del servizio: ListOperations
	individuazione del servizio: ListServices
	individuazione del servizio: RegisterInstance
	servicediscovery: spazio dei nomi UpdateHttp
	servicediscovery UpdateInstanceCustomHealth: stato
	individuazione del servizio: UpdatePrivate DnsNamespace
	individuazione del servizio: UpdatePublic DnsNamespace

Prefisso del servizio	Azioni
	individuazione del servizio: UpdateService
servicequotas	quote di servizio: AssociateService QuotaTemplate servicequotas: modello DeleteService QuotaIncrease RequestFrom quote di servizio: DisassociateService QuotaTemplate quote di servizio: GetAssociation ForService QuotaTemplate Quote di servizio: ottieni AWSDefaultServiceQuota quote di servizio: modifica GetRequested ServiceQuota quote di servizio: Quota GetService servicequotas: modello GetService QuotaIncrease RequestFrom Service Quotas: elenco AWSDefaultServiceQuotas quote di servizio: ListRequested ServiceQuota ChangeHistory quote di servizio: ListRequested ServiceQuota ChangeHistory ByQuota servicequotas: modello ListService QuotaIncrease RequestsIn servicequotas: quote ListService quote di servizio: ListServices servicequotas: modello PutService QuotaIncrease RequestInto quote di servizio: RequestService QuotaIncrease

Prefisso del servizio	Azioni
ses	usi: BatchGet MetricData usa: CloneReceipt RuleSet usi: CreateConfiguration Set ses: CreateConfiguration SetEvent Destinazione ses: CreateConfiguration SetTracking Opzioni usa: CreateContact ses: CreateContact Elenco ses: CreateCustom VerificationEmail Modello utilizza: CreateDedicated IpPool usa: CreateDeliverability TestReport ses: CreateEmail Identità usa: CreateEmail IdentityPolicy ses: CreateEmail modello usi: CreateImport Job ses: CreateReceipt Filtro ses: CreateReceipt Regola utilizza: CreateReceipt RuleSet usa: CreateTemplate usi: DeleteConfiguration Set ses: DeleteConfiguration SetEvent Destinazione ses: DeleteConfiguration SetTracking Opzioni

Prefisso del servizio	Azioni
	usa: DeleteContact
	ses: DeleteContact Elenco
	ses: DeleteCustom VerificationEmail Modello
	utilizza: DeleteDedicated IpPool
	ses: DeleteEmail Identità
	usa: DeleteEmail IdentityPolicy
	ses: DeleteEmail modello
	utilizza: DeleteIdentity
	ses: DeleteIdentity Politica
	usa: DeleteReceipt Filtro
	ses: DeleteReceipt Regola
	utilizza: DeleteReceipt RuleSet
	ses: DeleteSuppressed Destinazione
	usa: DeleteTemplate
	usa: DeleteVerified EmailAddress
	usi: DescribeActive ReceiptRule Set
	Usi: DescribeConfiguration Set
	ses: DescribeReceipt Regola
	utilizza: DescribeReceipt RuleSet
	usa: GetAccount
	usa: GetAccount SendingEnabled

Prefisso del servizio	Azioni
	ses: GetBlacklist Rapporti
	usa: GetConfiguration Set
	ses: GetConfiguration SetEvent Destinazioni
	usi: GetContact
	ses: GetContact Elenco
	ses: GetCustom VerificationEmail Modello
	usa: GetDedicated Ip
	utilizza: GetDedicated IpPool
	usi: GetDedicated Suggerimenti
	utilizza: GetDeliverability DashboardOptions
	usa: GetDeliverability TestReport
	usa: GetDomain DeliverabilityCampaign
	usa: GetDomain StatisticsReport
	ses: GetEmail Identità
	usa: GetEmail IdentityPolicies
	ses: GetEmail modello
	utilizza: GetIdentity DkimAttributes
	usa: GetIdentity MailFrom DomainAttributes
	usa: GetIdentity NotificationAttributes
	ses: GetIdentity Politiche
	utilizza: GetIdentity VerificationAttributes

Prefisso del servizio	Azioni
	usi: GetImport Job
	ses: GetMessage Approfondimenti
	usa: GetSend Quota
	ses: GetSend Statistiche
	ses: GetSuppressed Destinazione
	usa: GetTemplate
	usi: ListConfiguration set
	ses: ListContact elenchi
	utilizza: ListContacts
	ses: ListCustom VerificationEmail modelli
	utilizza: ListDedicated IpPools
	usa: ListDeliverability TestReports
	usa: ListDomain DeliverabilityCampaigns
	ses: ListEmail identità
	ses: Modelli ListEmail
	ses: ListExport Offerte di lavoro
	utilizza: ListIdentities
	ses: ListIdentity Politiche
	ses: ListImport Offerte di lavoro
	ses: ListReceipt Filtri
	utilizza: ListReceipt RuleSets

Prefisso del servizio	Azioni
	usa: ListRecommendations
	ses: ListSuppressed Destinazioni
	usi: ListTemplates
	usa: ListVerified EmailAddresses
	usa: PutAccount DedicatedIp WarmupAttributes
	ses: PutAccount Dettagli
	usi: PutAccount SendingAttributes
	usa: PutAccount SuppressionAttributes
	usa: PutAccount VdmAttributes
	ses: PutConfiguration SetDelivery Opzioni
	ses: PutConfiguration SetReputation Opzioni
	ses: PutConfiguration SetSending Opzioni
	ses: PutConfiguration SetSuppression Opzioni
	ses: PutConfiguration SetTracking Opzioni
	ses: PutConfiguration SetVdm Opzioni
	usi: PutDedicated IpIn Pool
	usi: PutDedicated IpPool ScalingAttributes
	ses: PutDedicated IpWarmup Attributi
	utilizza: PutDeliverability DashboardOption
	usa: PutEmail IdentityConfiguration SetAttributes
	ses: PutEmail IdentityDkim Attributi

Prefisso del servizio	Azioni
	utilizza: PutEmail IdentityDkim SigningAttributes
	ses: PutEmail IdentityFeedback Attributi
	utilizza: PutEmail IdentityMail FromAttributes
	ses: PutIdentity Politica
	ses: PutSuppressed Destinazione
	usa: ReorderReceipt RuleSet
	usa: SendBounce
	usa: SendCustom VerificationEmail
	usi: SetActive ReceiptRule Set
	utilizza: SetIdentity DkimEnabled
	ses: SetIdentity FeedbackForwarding abilitato
	utilizza: SetIdentity HeadersIn NotificationsEnabled
	ses: SetIdentity MailFrom Dominio
	usa: SetIdentity NotificationTopic
	usa: SetReceipt RulePosition
	usa: TestRender EmailTemplate
	ses: TestRender modello
	utilizza: UpdateAccount SendingEnabled
	ses: UpdateConfiguration SetEvent Destinazione
	usa: UpdateConfiguration SetReputation MetricsEnabled
	ses: UpdateConfiguration SetSending abilitato

Prefisso del servizio	Azioni
	ses: UpdateConfiguration SetTracking Opzioni usa: UpdateContact ses: UpdateContact Elenco ses: UpdateCustom VerificationEmail Modello utilizza: UpdateEmail IdentityPolicy ses: UpdateEmail modello ses: UpdateReceipt Regola utilizza: UpdateTemplate vedi: VerifyDomain Dikim ses: Identità VerifyDomain ses: VerifyEmail Indirizzo ses: VerifyEmail Identità

Prefisso del servizio	Azioni
shield	Shield: RT associato LogBucket shield: Controlla AssociateHealth scudo: AssociateProactive EngagementDetails scudo: CreateProtection scudo: CreateProtection Gruppo scudo: CreateSubscription scudo: DeleteProtection scudo: DeleteProtection Gruppo scudo: DeleteSubscription scudo: DescribeAttack shield: DescribeAttack Statistiche shield:DescribeDRTAccess shield: DescribeEmergency ContactSettings scudo: DescribeProtection scudo: DescribeProtection Gruppo scudo: DescribeSubscription scudo: DisableApplication LayerAutomatic risposta shield: DisableProactive Impegno Shield: RT dissociato LogBucket shield:DisassociateDRTRole shield: Controlla DisassociateHealth

Prefisso del servizio	Azioni
	shield: EnableApplication LayerAutomatic Risposta shield: EnableProactive Impegno scudo: GetSubscription Stato scudo: ListAttacks shield: ListProtection Gruppi scudo: ListProtections scudo: ListResources InProtection Gruppo shield: UpdateApplication LayerAutomatic Risposta scudo: UpdateEmergency ContactSettings scudo: UpdateProtection Gruppo scudo: UpdateSubscription

Prefisso del servizio	Azioni
signer	firmatario: autorizzazione AddProfile firmatario: Profilo CancelSigning firmatario: Job DescribeSigning firmatario: Status GetRevocation firmatario: Piattaforma GetSigning firmatario: Profilo GetSigning firmatario: autorizzazioni ListProfile firmatario: Jobs ListSigning firmatario: Piattaforme ListSigning firmatario: Profili ListSigning firmatario: Profilo PutSigning firmatario: autorizzazione RemoveProfile firmatario: RevokeSignature firmatario: Profilo RevokeSigning firmatario: SignPayload firmatario: Job StartSigning

Prefisso del servizio	Azioni
simspaceweaver	simspaceweaver: CreateSnapshot simspaceweaver: DeleteApp simspaceweaver: DeleteSimulation simspaceweaver: DescribeApp simspaceweaver: DescribeSimulation simspaceweaver: ListApps simspaceweaver: ListSimulations simspaceweaver: StartApp simspaceweaver: StartClock simspaceweaver: StartSimulation simspaceweaver: StopApp simspaceweaver: StopClock simspaceweaver: StopSimulation

Prefisso del servizio	Azioni
sms	sms: CreateApp
	sms: CreateReplication Job
	sms: DeleteApp
	sms: DeleteApp LaunchConfiguration
	sms: DeleteApp ReplicationConfiguration
	sms: DeleteApp ValidationConfiguration
	sms: DeleteReplication Job
	sms: DeleteServer Catalogo
	sms: DisassociateConnector
	sms: GenerateChange impostato
	sms: GenerateTemplate
	sms: GetApp
	sms: GetApp LaunchConfiguration
	sms: GetApp ReplicationConfiguration
	sms: GetApp ValidationConfiguration
	sms: GetApp ValidationOutput
	sms: GetConnectors
	sms: GetReplication Offerte di lavoro
	sms: GetReplication Esegue
	sms: GetServers
	sms: ImportApp Catalogo

Prefisso del servizio	Azioni
	sms: ImportServer Catalogo
	sms: LaunchApp
	sms: ListApps
	sms: NotifyApp ValidationOutput
	sms: PutApp LaunchConfiguration
	sms: PutApp ReplicationConfiguration
	sms: PutApp ValidationConfiguration
	sms: StartApp Replica
	sms: StartOn DemandApp Replica
	sms: StartOn DemandReplication Esegui
	sms: StopApp Replica
	sms: TerminateApp
	sms: UpdateApp
	sms: UpdateReplication Job

Prefisso del servizio	Azioni
sms-voice	sms-voice: Configurazione AssociateProtect sms-voice: Imposta CreateConfiguration sms-voice: Destinazione CreateConfiguration SetEvent sms-voice: Destinazione CreateEvent sms-voice: CreateOpt OutList messaggio vocale via sms: CreatePool sms-voice: Configurazione CreateProtect sms-voice: CreateRegistration sms-voice: Associazione CreateRegistration sms-voice: Allegato CreateRegistration sms-voice: versione CreateRegistration sms-voice: CreateVerified DestinationNumber sms-voice: Configurazione DeleteAccount DefaultProtect sms-voice: Imposta DeleteConfiguration sms-voice: Destinazione DeleteConfiguration SetEvent sms-voice: DeleteDefault MessageType SMS vocale: DeleteDefault SenderId sms-voice: Destinazione DeleteEvent sms-voice: DeleteKeyword SMS vocale: DeleteMedia MessageSpend LimitOverride messaggio vocale via sms: DeleteOpted OutNumber

Prefisso del servizio	Azioni
	SMS vocale: DeleteOpt OutList messaggio vocale via sms: DeletePool sms-voice: Configurazione DeleteProtect sms-voice: DeleteRegistration sms-voice: Allegato DeleteRegistration sms-voice: DeleteText MessageSpend LimitOverride messaggio vocale via sms: DeleteVerified DestinationNumber messaggio vocale via sms: DeleteVoice MessageSpend LimitOver ride sms-voice: Attributi DescribeAccount sms-voice: limiti DescribeAccount sms-voice: Imposta DescribeConfiguration sms-voice: DescribeKeywords messaggio vocale via sms: DescribeOpted OutNumbers SMS vocale: DescribeOpt OutLists sms-voice: numeri DescribePhone sms-voice: DescribePools sms-voice: configurazioni DescribeProtect sms-voice: Allegati DescribeRegistration sms-voice: DescribeRegistration FieldDefinitions SMS vocale: DescribeRegistration FieldValues

Prefisso del servizio	Azioni
	messaggio vocale via sms: DescribeRegistrations messaggio vocale via sms: DescribeRegistration SectionDefinitions SMS vocale: DescribeRegistration TypeDefinitions sms-voice: Versioni DescribeRegistration sms-voice: ID DescribeSender sms-voice: limiti DescribeSpend sms-voice: DescribeVerified DestinationNumbers sms-voice: Identità DisassociateOrigination sms-voice: Configurazione DisassociateProtect sms-voice: Versione DiscardRegistration sms-voice: Destinazioni GetConfiguration SetEvent sms-voice: GetProtect ConfigurationCountry RuleSet sms-voice: Imposta ListConfiguration sms-voice: ListPool OriginationIdentities sms-voice: Associazioni ListRegistration sms-voice: PutKeyword messaggio vocale via sms: PutOpted OutNumber sms-voice: numero ReleasePhone sms-voice: ID ReleaseSender sms-voice: numero RequestPhone sms-voice: ID RequestSender

Prefisso del servizio	Azioni
	sms-voice: Codice SendDestination NumberVerification sms-voice: Configurazione SetAccount DefaultProtect sms-voice: SetDefault MessageType SMS vocale: SetDefault SenderId messaggio vocale via sms: SetMedia MessageSpend LimitOverride messaggio vocale via sms: SetText MessageSpend LimitOverride messaggio vocale via sms: SetVoice MessageSpend LimitOverride sms-voice: versione SubmitRegistration sms-voice: Destinazione UpdateConfiguration SetEvent sms-voice: Destinazione UpdateEvent sms-voice: Numero UpdatePhone sms-voice: UpdatePool sms-voice: Configurazione UpdateProtect sms-voice: UpdateProtect ConfigurationCountry RuleSet sms-voice: ID UpdateSender

Prefisso del servizio	Azioni
snowball	palla di neve: CancelCluster
	palla di neve: CancelJob
	palla di neve: CreateAddress
	palla di neve: CreateCluster
	palla di neve: CreateJob
	palla di neve: CreateLong TermPricing
	palla di neve: CreateReturn ShippingLabel
	palla di neve: DescribeAddress
	palla di neve: DescribeAddresses
	palla di neve: DescribeCluster
	palla di neve: DescribeJob
	palla di neve: DescribeReturn ShippingLabel
	palla di neve: manifesto GetJob
	palla di neve: GetJob UnlockCode
	palla di neve: Utilizzo GetSnowball
	snowball: Aggiornamenti GetSoftware
	snowball: Offerte di lavoro ListCluster
	palla di neve: ListClusters
	palla di neve: immagini ListCompatible
	palla di neve: ListJobs
	palla di neve: ListLong TermPricing

Prefisso del servizio	Azioni
	snowball: sedi ListPickup snowball: Versioni ListService palla di neve: UpdateCluster palla di neve: UpdateJob palla di neve: UpdateJob ShipmentState palla di neve: UpdateLong TermPricing
sqs	seghe: AddPermission sq: CancelMessage MoveTask sq: CreateQueue sq: DeleteQueue sq: PurgeQueue sq: RemovePermission sqs: Attributi SetQueue

Prefisso del servizio	Azioni
ssm	ssm: Articolo AssociateOps ItemRelated
	ssm: CancelCommand
	ssm: CancelMaintenance WindowExecution
	ssm: CreateActivation
	ssm: CreateAssociation
	ssm: Batch CreateAssociation
	sms: CreateDocument
	ssm: Finestra CreateMaintenance
	ssm: Articolo CreateOps
	ssm: Metadati CreateOps
	ssm: Linea di base CreatePatch
	ssm: CreateResource DataSync
	ssm: DeleteActivation
	ssm: DeleteAssociation
	ssm: DeleteDocument
	ssm: DeleteInventory
	ssm: Finestra DeleteMaintenance
	ssm: Articolo DeleteOps
	ssm: Metadati DeleteOps
	ssm: DeleteParameter
	ssm: DeleteParameters

Prefisso del servizio	Azioni
	ssm: Linea di base DeletePatch
	ssm: DeleteResource DataSync
	ssm: Politica DeleteResource
	ssm: Istanza DeregisterManaged
	ssm: DeregisterPatch BaselineFor PatchGroup
	ssm: Finestra DeregisterTarget FromMaintenance
	ssm: Finestra DeregisterTask FromMaintenance
	ssm: DescribeActivations
	ssm: DescribeAssociation
	ssm: Esecuzioni DescribeAssociation
	ssm: DescribeAssociation ExecutionTargets
	ssm: Esecuzioni DescribeAutomation
	ssm: DescribeAutomation StepExecutions
	ssm: Patch DescribeAvailable
	ssm: DescribeDocument
	ssm: Parametri DescribeDocument
	ssm: Autorizzazione DescribeDocument
	ssm: DescribeEffective InstanceAssociations
	ssm: DescribeEffective PatchesFor PatchBaseline
	ssm: DescribeInstance AssociationsStatus
	ssm: Informazioni DescribeInstance

Prefisso del servizio	Azioni
	ssm: Patch DescribeInstance
	ssm: DescribeInstance PatchStates
	ssm: Gruppo DescribeInstance PatchStates ForPatch
	ssm: Proprietà DescribeInstance
	ssm: Eliminazioni DescribeInventory
	ssm: DescribeMaintenance WindowExecutions
	ssm: DescribeMaintenance WindowExecution TaskInvocations
	ssm: Attività DescribeMaintenance WindowExecution
	ssm: Windows DescribeMaintenance
	ssm: DescribeMaintenance WindowSchedule
	ssm: Obiettivo DescribeMaintenance WindowsFor
	ssm: DescribeMaintenance WindowTargets
	ssm: DescribeMaintenance WindowTasks
	ssm: Articoli DescribeOps
	ssm: DescribeParameters
	ssm: Linee di base DescribePatch
	ssm: Gruppi DescribePatch
	ssm: DescribePatch GroupState
	ssm: Proprietà DescribePatch
	ssm: DescribeSessions
	ssm: Articolo DisassociateOps ItemRelated

Prefisso del servizio	Azioni
	ssm: Esecuzione GetAutomation
	ssm: Stato GetCalendar
	ssm: Invocazione GetCommand
	ssm: Stato GetConnection
	ssm: GetDefault PatchBaseline
	ssm: GetDeployable PatchSnapshot ForInstance
	ssm: GetDocument
	ssm: GetInventory
	ssm: schema GetInventory
	ssm: Finestra GetMaintenance
	ssm: GetMaintenance WindowExecution
	ssm: Attività GetMaintenance WindowExecution
	ssm: GetMaintenance WindowExecution TaskInvocation
	ssm: GetMaintenance WindowTask
	ssm: Articolo GetOps
	ssm: Metadati GetOps
	ssm: Riepilogo GetOps
	ssm: GetParameter
	ssm: Storia GetParameter
	ssm: GetParameters
	ssm: GetParameters ByPath

Prefisso del servizio	Azioni
	ssm: Linea di base GetPatch
	ssm: GetPatch BaselineFor PatchGroup
	ssm: Politiche GetResource
	ssm: Impostazione GetService
	ssm: versione LabelParameter
	ssm: ListAssociations
	ssm: Versioni ListAssociation
	ssm: invocazioni ListCommand
	ssm: ListCommands
	ssm: Articoli ListCompliance
	ssm: Riassunti ListCompliance
	ssm: ListDocument MetadataHistory
	ssm: ListDocuments
	ssm: Versioni ListDocument
	ssm: Associazioni ListInstance
	ssm: Iscrizioni ListInventory
	ssm: ListOps ItemEvents
	ssm: Articoli ListOps ItemRelated
	ssm: Metadati ListOps
	ssm: ListResource ComplianceSummaries
	ssm: ListResource DataSync

Prefisso del servizio	Azioni
	ssm: Autorizzazione ModifyDocument
	ssm: Articoli PutCompliance
	ssm: PutInventory
	ssm: PutParameter
	ssm: Politica PutResource
	ssm: RegisterDefault PatchBaseline
	ssm: Istanza RegisterManaged
	ssm: RegisterPatch BaselineFor PatchGroup
	ssm: Finestra RegisterTarget WithMaintenance
	ssm: Finestra RegisterTask WithMaintenance
	ssm: Impostazione ResetService
	ssm: ResumeSession
	ssm: Segnale SendAutomation
	ssm: SendCommand
	ssm: Una volta StartAssociations
	ssm: Esecuzione StartAutomation
	ssm: StartChange RequestExecution
	ssm: StartSession
	ssm: Esecuzione StopAutomation
	ssm: TerminateSession
	ssm: versione UnlabelParameter

Prefisso del servizio	Azioni
	ssm: UpdateAssociation
	ssm: Stato UpdateAssociation
	ssm: UpdateDocument
	ssm: UpdateDocument DefaultVersion
	ssm: Metadati UpdateDocument
	ssm: Informazioni UpdateInstance
	ssm: Finestra UpdateMaintenance
	ssm: UpdateMaintenance WindowTarget
	ssm: UpdateMaintenance WindowTask
	ssm: UpdateManaged InstanceRole
	ssm: Articolo UpdateOps
	ssm: Metadati UpdateOps
	ssm: Linea di base UpdatePatch
	ssm: UpdateResource DataSync
	ssm: Impostazione UpdateService

Prefisso del servizio	Azioni
ssm-incidents	ssm-incidenti: BatchGet IncidentFindings ssm-incidents: impostato CreateReplication ssm-incidents: CreateResponse Pianifica ssm-incidents: CreateTimeline Evento ssm-incidents: DeleteIncident Registra ssm-incidents: DeleteReplication impostato ssm-incidents: DeleteResource politica ssm-incidents: DeleteResponse Piano ssm-incidents: DeleteTimeline Evento ssm-incidents: GetIncident Registra ssm-incidents: GetReplication impostato ssm-incidents: GetResource politiche ssm-incidents: GetResponse Piano ssm-incidents: GetTimeline Evento ssm-incidents: ListIncident risultati ssm-incidents: ListIncident record ssm-incidents: ListRelated Articoli ssm-incidents: ListReplication insiemi ssm-incidents: ListResponse Piani ssm-incidents: ListTimeline Eventi PutResourcessm-incidents: Politica

Prefisso del servizio	Azioni
	ssm-incidents: StartIncident ssm-incidents: protezione UpdateDeletion ssm-incidents: UpdateIncident Registra ssm-incidents: UpdateRelated articoli ssm-incidents: UpdateReplication impostato ssm-incidents: UpdateResponse Pianifica ssm-incidents: UpdateTimeline Evento

Prefisso del servizio	Azioni
ssm-sap	ssm-sap: BackupDatabase
	ssm-sap: Autorizzazione DeleteResource
	ssm-sap: DeregisterApplication
	ssm-sap: GetApplication
	ssm-sap: GetComponent
	ssm-sap: GetDatabase
	ssm-sap: GetOperation
	ssm-sap: Autorizzazione GetResource
	ssm-sap: ListApplications
	ssm-sap: ListComponents
	ssm-sap: ListDatabases
	ssm-sap: Eventi ListOperation
	ssm-sap: ListOperations
	ssm-sap: Autorizzazione PutResource
	ssm-sap: RegisterApplication
	ssm-sap: RestoreDatabase
	ssm-sap: StartApplication
	ssm-sap: Aggiorna StartApplication
	ssm-sap: StopApplication
	ssm-sap: Impostazioni UpdateApplication
	Ssm-sap: aggiorna Hana BackupSettings

Prefisso del servizio	Azioni
states	afferma: CreateActivity stati: CreateState Macchina stati: CreateState MachineAlias stati: DeleteActivity stati: DeleteState Macchina stati: DeleteState MachineAlias stati: DeleteState MachineVersion stati: DescribeActivity stati: DescribeExecution stati: DescribeMap Esegui stati: DescribeState Macchina stati: DescribeState MachineAlias stati: DescribeState MachineFor esecuzione stati: GetExecution Storia stati: ListActivities stati: ListExecutions stati: ListMap Esegue stati: ListState MachineAliases stati: ListState Macchine stati: ListState MachineVersions stati: SendTask fallimento

Prefisso del servizio	Azioni
	stati: SendTask Heartbeat stati: Successo SendTask stati: StartExecution stati: StopExecution stati: UpdateMap Esegui stati: UpdateState Macchina stati: UpdateState MachineAlias stati: ValidateState MachineDefinition
sts	set: AssumeRole sts: AssumeRole con SAML sts: Identità AssumeRole WithWeb sts: DecodeAuthorization Messaggio sts: GetAccess KeyInfo sts: GetCaller Identità sts: GetFederation Token sts: GetSession Token

Prefisso del servizio	Azioni
swf	swf: tipo DeprecateActivity swf: DeprecateDomain swf: tipo DeprecateWorkflow swf: tipo DescribeActivity swf: DescribeDomain swf: tipo DescribeWorkflow swf: tipi ListActivity swf: ListDomains swf: tipi ListWorkflow swf: Tipo RegisterActivity swf: RegisterDomain swf: tipo RegisterWorkflow swf: tipo UndeprecateActivity swf: UndeprecateDomain swf: tipo UndeprecateWorkflow

Prefisso del servizio	Azioni
synthetics	sintetici: AssociateResource
	sintetici: CreateCanary
	sintetici: CreateGroup
	sintetici: DeleteCanary
	sintetici: DeleteGroup
	sintetici: DescribeCanaries
	sintetici: DescribeCanaries LastRun
	sintetici: versioni DescribeRuntime
	sintetici: DisassociateResource
	sintetici: GetCanary
	sintetici: Runs GetCanary
	sintetici: GetGroup
	sintetici: gruppi ListAssociated
	sintetici: risorse ListGroup
	sintetici: ListGroups
	sintetici: StartCanary
	sintetici: StopCanary
	sintetici: UpdateCanary

Prefisso del servizio	Azioni
tag	tag: Creazione DescribeReport tag: GetCompliance Riepilogo etichetta: GetResources tag: StartReport Creazione

Prefisso del servizio	Azioni
textract	estratto: AnalyzeDocument
	estratto: AnalyzeExpense
	textract:AnalyzeID
	estratto: CreateAdapter
	textract: Versione CreateAdapter
	estratto: DeleteAdapter
	textract: Versione DeleteAdapter
	textract: Testo DetectDocument
	estratto: GetAdapter
	textract: Versione GetAdapter
	textract: Analisi GetDocument
	estratto: GetDocument TextDetection
	textract: Analisi GetExpense
	textract: Analisi GetLending
	estratto: GetLending AnalysisSummary
	estratto: ListAdapters
	textract: Versioni ListAdapter
	textract: Analisi StartDocument
	estratto: StartDocument TextDetection
	textract: Analisi StartExpense
	textract: Analisi StartLending

Prefisso del servizio	Azioni
	estratto: UpdateAdapter

Prefisso del servizio	Azioni
timestream	flusso temporale: CancelQuery flusso temporale: CreateDatabase flusso temporale: Interrogazione CreateScheduled flusso temporale: CreateTable flusso temporale: DeleteDatabase flusso temporale: Interrogazione DeleteScheduled flusso temporale: DeleteTable timestream: impostazioni DescribeAccount flusso temporale: DescribeDatabase flusso temporale: Interrogazione DescribeScheduled flusso temporale: DescribeTable flusso temporale: Interrogazione ExecuteScheduled flusso temporale: ListBatch LoadTasks flusso temporale: ListDatabases timestream: domande ListScheduled flusso temporale: ListTables flusso temporale: PrepareQuery timestream: impostazioni UpdateAccount flusso temporale: UpdateDatabase flusso temporale: Interrogazione UpdateScheduled flusso temporale: UpdateTable

Prefisso del servizio	Azioni
tnb	tb: CancelSol NetworkOperation
	tnb: CreateSol FunctionPackage
	tnb: CreateSol NetworkInstance
	tnb: CreateSol NetworkPackage
	tnb: DeleteSol FunctionPackage
	tnb: DeleteSol NetworkInstance
	tnb: DeleteSol NetworkPackage
	tnb: GetSol FunctionInstance
	tnb: GetSol FunctionPackage
	tnb: Contenuto GetSol FunctionPackage
	tnb: descrittore GetSol FunctionPackage
	tnb: GetSol NetworkInstance
	tnb: GetSol NetworkOperation
	tnb: GetSol NetworkPackage
	tnb: Contenuto GetSol NetworkPackage
	tnb: descrittore GetSol NetworkPackage
	tnb: InstantiateSol NetworkInstance
	tnb: ListSol FunctionInstances
	tnb: ListSol FunctionPackages
	tnb: ListSol NetworkInstances
	tnb: ListSol NetworkOperations

Prefisso del servizio	Azioni
	tnb: ListSol NetworkPackages
	tnb: Contenuto PutSol FunctionPackage
	tnb: Contenuto PutSol NetworkPackage
	tnb: TerminateSol NetworkInstance
	tnb: UpdateSol FunctionPackage
	tnb: UpdateSol NetworkInstance
	tnb: UpdateSol NetworkPackage
	tnb: Contenuto ValidateSol FunctionPackage
	tnb: Contenuto ValidateSol NetworkPackage

Prefisso del servizio	Azioni
transcribe	trascrivere: CreateCall AnalyticsCategory trascrivere: modello CreateLanguage trascrivere: Vocabolario CreateMedical trascrivere: CreateVocabulary trascrivere: filtro CreateVocabulary trascrivere: DeleteCall AnalyticsCategory trascrivere: DeleteCall AnalyticsJob trascrivere: modello DeleteLanguage trascrivere: DeleteMedical ScribeJob trascrivere: DeleteMedical TranscriptionJob trascrivere: vocabolario DeleteMedical trascrivere: Job DeleteTranscription trascrivere: DeleteVocabulary trascrivere: filtro DeleteVocabulary trascrivere: Modello DescribeLanguage trascrivere: GetCall AnalyticsCategory trascrivere: GetCall AnalyticsJob trascrivere: GetMedical ScribeJob trascrivere: GetMedical TranscriptionJob trascrivere: vocabolario GetMedical trascrivere: Job GetTranscription

Prefisso del servizio	Azioni
	trascrivere: GetVocabulary trascrivere: filtro GetVocabulary trascrivere: ListCall AnalyticsCategories trascrivere: ListCall AnalyticsJobs trascrivere: Modelli ListLanguage trascrivere: ListMedical ScribeJobs trascrivere: ListMedical TranscriptionJobs trascrivere: vocabolari ListMedical trascrivere: ListTranscription Offerte di lavoro trascrivere: ListVocabularies trascrivere: filtri ListVocabulary trascrivere: StartCall AnalyticsJob trascrivere: Trascrizione StartCall AnalyticsStream trascrivere: StartCall AnalyticsStream TranscriptionWeb Socket trascrivere: StartMedical ScribeJob trascrivere: StartMedical StreamTranscription trascrivere: StartMedical StreamTranscription WebSocket trascrivere: StartMedical TranscriptionJob trascrivere: Trascrizione StartStream trascrivere: StartStream TranscriptionWeb Socket trascrivere: Job StartTranscription

Prefisso del servizio	Azioni
	trascrivere: UpdateCall AnalyticsCategory trascrivere: vocabolario UpdateMedical trascrivere: UpdateVocabulary trascrivere: filtro UpdateVocabulary

Prefisso del servizio	Azioni
transfer	trasferimento: CreateAccess
	trasferimento: CreateAgreement
	trasferimento: CreateConnector
	trasferimento: CreateProfile
	trasferimento: CreateServer
	trasferimento: CreateUser
	trasferimento: CreateWorkflow
	trasferimento: DeleteAccess
	trasferimento: DeleteAgreement
	trasferimento: DeleteCertificate
	trasferimento: DeleteConnector
	trasferimento: DeleteHost chiave
	trasferimento: DeleteProfile
	trasferimento: DeleteServer
	trasferimento: DeleteSsh PublicKey
	trasferimento: DeleteUser
	trasferimento: DeleteWorkflow
	trasferimento: DescribeAccess
	trasferimento: DescribeAgreement
	trasferimento: DescribeCertificate
	trasferimento: DescribeConnector

Prefisso del servizio	Azioni
	trasferimento: DescribeExecution
	trasferimento: DescribeHost chiave
	trasferimento: DescribeProfile
	trasferimento: DescribeSecurity politica
	trasferimento: DescribeServer
	trasferimento: DescribeUser
	trasferimento: DescribeWorkflow
	trasferimento: ImportCertificate
	trasferimento: ImportHost chiave
	trasferimento: ImportSsh PublicKey
	trasferimento: ListAccesses
	trasferimento: ListCertificates
	trasferimento: ListConnectors
	trasferimento: ListExecutions
	trasferimento: ListHost chiavi
	trasferimento: ListProfiles
	trasferimento: ListSecurity politiche
	trasferimento: ListServers
	trasferimento: ListUsers
	trasferimento: ListWorkflows
	trasferimento: SendWorkflow StepState

Prefisso del servizio	Azioni
	trasferimento: StartDirectory elenco
	trasferimento: StartFile Trasferimento
	trasferimento: StartServer
	trasferimento: StopServer
	trasferimento: TestConnection
	trasferimento: TestIdentity fornitore
	trasferimento: UpdateAccess
	trasferimento: UpdateAgreement
	trasferimento: UpdateCertificate
	trasferimento: UpdateConnector
	trasferimento: UpdateHost chiave
	trasferimento: UpdateProfile
	trasferimento: UpdateServer
	trasferimento: UpdateUser

Prefisso del servizio	Azioni
translate	tradurre: CreateParallel Dati tradurre: DeleteParallel Dati tradurre: DeleteTerminology tradurre: DescribeText TranslationJob tradurre: GetParallel Dati tradurre: GetTerminology tradurre: ImportTerminology tradurre: ListLanguages tradurre: ListParallel Dati tradurre: ListTerminologies tradurre: ListText TranslationJobs tradurre: StartText TranslationJob tradurre: StopText TranslationJob tradurre: TranslateDocument tradurre: TranslateText tradurre: UpdateParallel Dati

Prefisso del servizio	Azioni
voiceid	voiceid: AssociateFraudster
	ID vocale: CreateDomain
	ID vocale: CreateWatchlist
	ID vocale: DeleteDomain
	ID vocale: DeleteFraudster
	ID vocale: DeleteSpeaker
	ID vocale: DeleteWatchlist
	ID vocale: DescribeDomain
	ID vocale: DescribeFraudster
	identificatore vocale: DescribeFraudster RegistrationJob
	identificatore vocale: DescribeSpeaker
	identificatore vocale: DescribeSpeaker EnrollmentJob
	identificatore vocale: DescribeWatchlist
	ID vocale: DisassociateFraudster
	ID vocale: EvaluateSession
	ID vocale: ListDomains
	identificatore vocale: ListFraudster RegistrationJobs
	identificatore vocale: ListFraudsters
	identificatore vocale: ListSpeaker EnrollmentJobs
	identificatore vocale: ListSpeakers
	ID vocale: ListWatchlists

Prefisso del servizio	Azioni
	voiceid: altoparlante OptOut identificatore vocale: StartFraudster RegistrationJob identificatore vocale: StartSpeaker EnrollmentJob identificatore vocale: UpdateDomain ID vocale: UpdateWatchlist

Prefisso del servizio	Azioni
vpc-lattice	reticolo vpc: CreateAccess LogSubscription reticolo vpc: CreateListener reticolo vpc: CreateRule reticolo vpc: CreateService vpc-lattice: CreateService Rete vpc-lattice: CreateService NetworkService Associazione vpc-lattice: CreateService NetworkVpc Associazione vpc-lattice: CreateTarget Gruppo reticolo vpc: DeleteAccess LogSubscription vpc-lattice: politica DeleteAuth reticolo vpc: DeleteListener vpc-lattice: politica DeleteResource reticolo vpc: DeleteRule reticolo vpc: DeleteService vpc-lattice: DeleteService Rete vpc-lattice: DeleteService NetworkService Associazione vpc-lattice: DeleteService NetworkVpc Associazione vpc-lattice: DeleteTarget Gruppo reticolo vpc: DeregisterTargets reticolo vpc: GetAccess LogSubscription vpc-lattice: politica GetAuth

Prefisso del servizio	Azioni
	reticolo vpc: GetListener vpc-lattice: politica GetResource reticolo vpc: GetRule reticolo vpc: GetService vpc-lattice: GetService Rete vpc-lattice: GetService NetworkService Associazione vpc-lattice: GetService NetworkVpc Associazione vpc-lattice: GetTarget Gruppo reticolo vpc: ListAccess LogSubscriptions reticolo vpc: ListListeners reticolo vpc: ListRules vpc-lattice: ListService Reti vpc-lattice: ListService NetworkService Associazioni vpc-lattice: ListService NetworkVpc Associazioni vpc-reticolo: ListServices vpc-lattice: Gruppi ListTarget reticolo vpc: ListTargets vpc-lattice: politica PutAuth vpc-lattice: PutResource politica reticolo vpc: RegisterTargets reticolo vpc: UpdateAccess LogSubscription

Prefisso del servizio	Azioni
	reticolo vpc: UpdateListener reticolo vpc: UpdateRule reticolo vpc: UpdateService vpc-lattice: UpdateService Rete vpc-lattice: UpdateService NetworkVpc Associazione vpc-lattice: UpdateTarget Gruppo

Prefisso del servizio	Azioni
wafv2	wafv2: AssociateWeb ACL
	wafv2: CheckCapacity
	wafv2:CreateAPIKey
	wafv2:CreateIPSet
	wafv2: CreateRegex PatternSet
	wafv2: Gruppo CreateRule
	wafv2: ACL CreateWeb
	WAFv2: elimina la chiave API
	wafv2: Gruppi DeleteFirewall ManagerRule
	wafv2:DeleteIPSet
	wafv2: Configurazione DeleteLogging
	wafv2: politica DeletePermission
	wafv2: DeleteRegex PatternSet
	wafv2: Gruppo DeleteRule
	wafv2: ACL DeleteWeb
	wafv2: DescribeAll ManagedProducts
	wafv2: Fornitore DescribeManaged ProductsBy
	wafv2: DescribeManaged RuleGroup
	wafv2: ACL DisassociateWeb
	wafv2: GenerateMobile SdkRelease URL
	wafv2: API Key GetDecrypted

Prefisso del servizio	Azioni
	wafv2:GetIPSet
	wafv2: GetLogging Configurazione
	wafv2: GetManaged RuleSet
	wafv2: GetMobile SdkRelease
	wafv2: Politica GetPermission
	wafv2: GetRate BasedStatement ManagedKeys
	wafv2: GetRegex PatternSet
	wafv2: Gruppo GetRule
	wafv2: Richieste GetSampled
	wafv2: ACL GetWeb ForResource
	wafv2:ListAPIKeys
	wafv2: ListAvailable ManagedRule Gruppi
	wafv2: ListAvailable ManagedRule GroupVersions
	wafv2:ListIPSets
	wafv2: configurazioni ListLogging
	wafv2: ListManaged RuleSets
	wafv2: ListMobile SdkReleases
	wafv2: ListRegex PatternSets
	wafv2: ACL ListResources ForWeb
	wafv2: ListRule Gruppi
	wafv2: ACL ListWeb

Prefisso del servizio	Azioni
	wafv2: PutLogging Configurazione
	wafv2: versioni PutManaged RuleSet
	wafv2: Politica PutPermission
	wafv2:UpdateIPSet
	wafv2: Data UpdateManaged RuleSet VersionExpiry
	wafv2: UpdateRegex PatternSet
	wafv2: Gruppo UpdateRule
	wafv2: ACL UpdateWeb

Prefisso del servizio	Azioni
wellarchitected	ben architettato: AssociateLenses ben architettato: AssociateProfiles wellarchitected: Condividi CreateLens wellarchitected: versione CreateLens ben architettato: CreateMilestone ben architettato: CreateProfile wellarchitected: Condividi CreateProfile wellarchitected: Modello CreateReview ben architettato: CreateWorkload wellarchitected: Condividi CreateWorkload ben architettato: DeleteLens wellarchitected: Condividi DeleteLens ben architettato: DeleteProfile wellarchitected: Condividi DeleteProfile wellarchitected: Modello DeleteReview wellarchitected: Condividi DeleteTemplate ben architettato: DeleteWorkload wellarchitected: Condividi DeleteWorkload ben architettato: DisassociateLenses ben architettato: DisassociateProfiles ben architettato: ExportLens

Prefisso del servizio	Azioni
	ben architettato: GetAnswer ben architettato: rapporto GetConsolidated wellarchitected: Impostazioni GetGlobal ben architettato: GetLens wellarchitected: Recensione GetLens ben architettato: GetLens ReviewReport ben architettato: GetLens VersionDifference ben architettato: GetMilestone ben architettato: GetProfile wellarchitected: Modello GetProfile wellarchitected: Modello GetReview ben architettato: GetReview TemplateAnswer wellarchitected: Recensione GetReview TemplateLens ben architettato: GetWorkload ben architettato: ImportLens ben architettato: ListAnswers ben architettato: Dettagli ListCheck wellarchitected: Riassunti ListCheck ben architettato: ListLenses ben architettato: ListLens ReviewImprovements wellarchitected: recensioni ListLens

Prefisso del servizio	Azioni
	wellarchitected: Shares ListLens ben architettato: ListMilestones ben architettato: ListNotifications wellarchitected: notifiche ListProfile ben architettato: ListProfiles ben architettato: Shares ListProfile ben architettato: ListReview TemplateAnswers wellarchitected: Modelli ListReview wellarchitected: Inviti ListShare ben architettato: Shares ListTemplate ben architettato: ListWorkloads ben architettato: Shares ListWorkload ben architettato: UpdateAnswer wellarchitected: Impostazioni UpdateGlobal ben architettato: UpdateIntegration wellarchitected: Recensione UpdateLens ben architettato: UpdateProfile wellarchitected: Modello UpdateReview wellarchitected: Recensione UpdateReview TemplateLens wellarchitected: Invito UpdateShare ben architettato: UpdateWorkload

Prefisso del servizio	Azioni
	wellarchitected: Condividi UpdateWorkload wellarchitected: Recensione UpgradeLens wellarchitected: versione UpgradeProfile wellarchitected: Recensione UpgradeReview TemplateLens

Prefisso del servizio	Azioni
wisdom	saggezza: CreateAssistant
	saggezza: CreateAssistant Associazione
	saggezza: CreateContent
	saggezza: CreateKnowledge Base
	saggezza: CreateQuick Risposta
	saggezza: CreateSession
	saggezza: DeleteAssistant
	saggezza: DeleteAssistant Associazione
	saggezza: DeleteContent
	wisdom: DeleteImport Job
	saggezza: DeleteKnowledge Base
	saggezza: DeleteQuick Risposta
	saggezza: GetAssistant
	saggezza: GetAssistant Associazione
	saggezza: GetContent
	saggezza: GetContent Riepilogo
	wisdom: GetImport Job
	saggezza: GetKnowledge Base
	saggezza: GetRecommendations
	saggezza: GetSession
	saggezza: ListAssistant Associazioni

Prefisso del servizio	Azioni
	saggezza: ListAssistants
	saggezza: ListContents
	saggezza: ListImport Offerte di lavoro
	saggezza: ListKnowledge basi
	saggezza: ListQuick Risposte
	saggezza: NotifyRecommendations ricevuta
	saggezza: QueryAssistant
	saggezza: RemoveKnowledge BaseTemplate Uri
	saggezza: SearchContent
	saggezza: SearchQuick Risposte
	saggezza: SearchSessions
	wisdom: StartContent Carica
	wisdom: StartImport Job
	saggezza: UpdateContent
	saggezza: UpdateKnowledge BaseTemplate Uri
	saggezza: UpdateQuick Risposta

Prefisso del servizio	Azioni
worklink	collegamento di lavoro: AssociateDomain collegamento di lavoro: AssociateWebsite AuthorizationProvider collegamento di lavoro: AssociateWebsite CertificateAuthority collegamento di lavoro: CreateFleet collegamento di lavoro: DeleteFleet collegamento di lavoro: DescribeAudit StreamConfiguration collegamento di lavoro: DescribeCompany NetworkConfiguration collegamento di lavoro: DescribeDevice collegamento di lavoro: DescribeDevice PolicyConfiguration collegamento di lavoro: DescribeDomain worklink: Metadati DescribeFleet collegamento di lavoro: DescribeIdentity ProviderConfiguration collegamento di lavoro: DescribeWebsite CertificateAuthority collegamento di lavoro: DisassociateDomain collegamento di lavoro: DisassociateWebsite AuthorizationProvider collegamento di lavoro: DisassociateWebsite CertificateAuthority collegamento di lavoro: ListDevices collegamento di lavoro: ListDomains collegamento di lavoro: ListFleets collegamento di lavoro: ListWebsite AuthorizationProviders collegamento di lavoro: ListWebsite CertificateAuthorities

Prefisso del servizio	Azioni
	collegamento di lavoro: Access RestoreDomain worklink: Access RevokeDomain worklink: Utente SignOut collegamento di lavoro: UpdateAudit StreamConfiguration collegamento di lavoro: UpdateCompany NetworkConfiguration collegamento di lavoro: UpdateDevice PolicyConfiguration worklink: Metadati UpdateDomain worklink: Metadati UpdateFleet collegamento di lavoro: UpdateIdentity ProviderConfiguration

Prefisso del servizio	Azioni
workspace	spazi di lavoro: AcceptAccount LinkInvitation
	spazi di lavoro: alias AssociateConnection
	spazi di lavoro: gruppi AssociateIp
	spazi di lavoro: Applicazione AssociateWorkspace
	spazi di lavoro: Immagine CopyWorkspace
	spazi di lavoro: In CreateConnect ClientAdd
	spazi di lavoro: Alias CreateConnection
	spazi di lavoro: Gruppo CreateIp
	spazi di lavoro: Spazi di lavoro CreateStandby
	spazi di lavoro: CreateUpdated WorkspaceImage
	spazi di lavoro: pacchetto CreateWorkspace
	spazi di lavoro: immagine CreateWorkspace
	spazi di lavoro: CreateWorkspaces
	spazi di lavoro: branding DeleteClient
	spazi di lavoro: In DeleteConnect ClientAdd
	spazi di lavoro: Alias DeleteConnection
	spazi di lavoro: Gruppo DeleteIp
	spazi di lavoro: pacchetto DeleteWorkspace
	spazi di lavoro: immagine DeleteWorkspace
	spazi di lavoro: applicazioni DeployWorkspace
	spazi di lavoro: Directory DeregisterWorkspace

Prefisso del servizio	Azioni
	spazi di lavoro: DescribeAccount
	spazi di lavoro: modifiche DescribeAccount
	spazi di lavoro: associazioni DescribeApplication
	spazi di lavoro: DescribeApplications
	spazi di lavoro: associazioni DescribeBundle
	spazi di lavoro: Branding DescribeClient
	spazi di lavoro: Proprietà DescribeClient
	spazi di lavoro: Ins DescribeConnect ClientAdd
	workspaces: alias DescribeConnection
	spazi di lavoro: DescribeConnection AliasPermissions
	spazi di lavoro: associazioni DescribeImage
	spazi di lavoro: gruppi DescribeIp
	spazi di lavoro: associazioni DescribeWorkspace
	spazi di lavoro: pacchetti DescribeWorkspace
	spazi di lavoro: elenchi DescribeWorkspace
	spazi di lavoro: DescribeWorkspace ImagePermissions
	spazi di lavoro: DescribeWorkspaces
	spazi di lavoro: DescribeWorkspaces ConnectionStatus
	aree di lavoro: istantanee DescribeWorkspace
	spazi di lavoro: Alias DisassociateConnection
	spazi di lavoro: gruppi DisassociateIp

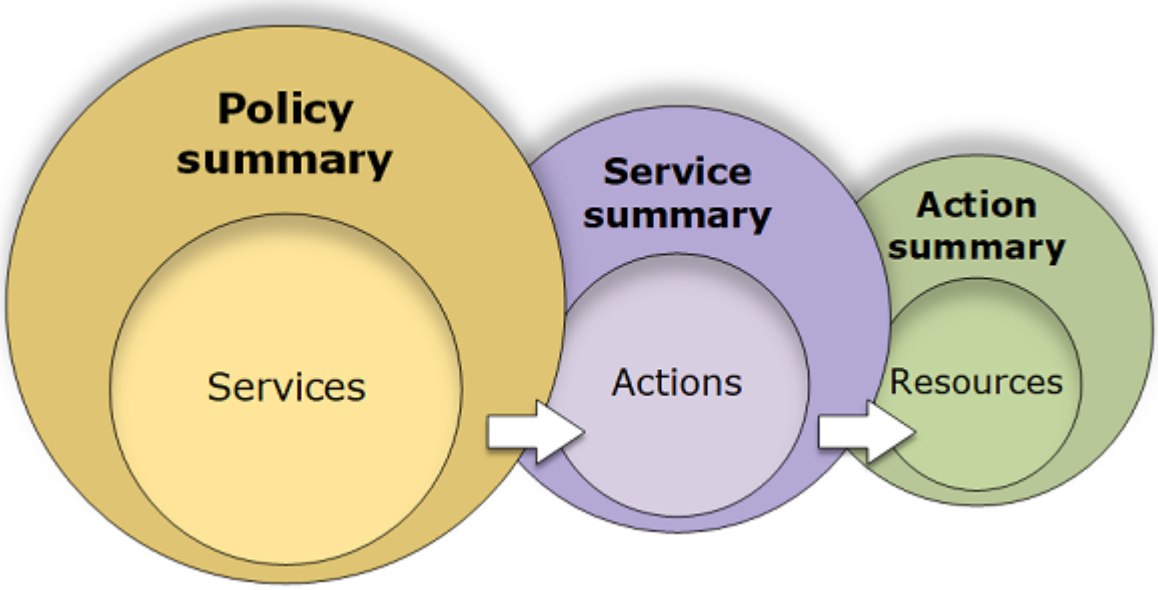
Prefisso del servizio	Azioni
	spazi di lavoro: Applicazione DisassociateWorkspace
	spazi di lavoro: Link GetAccount
	spazi di lavoro: Branding ImportClient
	spazi di lavoro: immagine ImportWorkspace
	spazi di lavoro: collegamenti ListAccount
	spazi di lavoro: intervalli ListAvailable ManagementCidr
	spazi di lavoro: MigrateWorkspace
	spazi di lavoro: ModifyAccount
	spazi di lavoro: proprietà ModifyCertificate BasedAuth
	spazi di lavoro: Proprietà ModifyClient
	spazi di lavoro: Proprietà ModifySaml
	spazi di lavoro: autorizzazioni ModifySelfservice
	spazi di lavoro: ModifyWorkspace AccessProperties
	spazi di lavoro: ModifyWorkspace CreationProperties
	spazi di lavoro: proprietà ModifyWorkspace
	spazi di lavoro: Stato ModifyWorkspace
	spazi di lavoro: RebootWorkspaces
	spazi di lavoro: RebuildWorkspaces
	spazi di lavoro: Directory RegisterWorkspace
	spazi di lavoro: RejectAccount LinkInvitation
	spazi di lavoro: RestoreWorkspace

Prefisso del servizio	Azioni
	spazi di lavoro: StartWorkspaces
	spazi di lavoro: StopWorkspaces
	spazi di lavoro: TerminateWorkspaces
	spazi di lavoro: In UpdateConnect ClientAdd
	spazi di lavoro: UpdateConnection AliasPermission
	spazi di lavoro: pacchetto UpdateWorkspace
	spazi di lavoro: UpdateWorkspace ImagePermission

Prefisso del servizio	Azioni
xray	radiografia: CreateGroup radiografia: regola CreateSampling Radiografia: DeleteGroup xray: politica DeleteResource xray: Regola DeleteSampling xray: Config GetEncryption radiografia: GetGroup radiografia: GetGroups radiografia: GetInsight xray: Eventi GetInsight radiografia: GetInsight ImpactGraph xray: riassunti GetInsight xray: Regole GetSampling xray: Politiche ListResource xray: Config PutEncryption xray: Politica PutResource Radiografia: UpdateGroup radiografia: regola UpdateSampling

Informazioni sulle autorizzazioni concesse da una policy

La console IAM include tabelle di riepilogo di policy che descrivono il livello di accesso, le risorse e le condizioni concesse o rifiutate per ciascun servizio in una policy. Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella di riepilogo della policy include un elenco di servizi. Scegli un servizio per visualizzare il riepilogo del servizio. Questa tabella include un elenco delle operazioni e le autorizzazioni associate per il servizio scelto. È possibile scegliere un'operazione dalla tabella per visualizzare il riepilogo dell'operazione. Questa tabella include un elenco di risorse e condizioni per l'operazione scelta.



Puoi visualizzare i riepiloghi delle policy nella pagina Users (Utenti) o Roles (Ruoli) per tutte le policy (gestite e inline) collegate a tale utente. e visualizzare i riepiloghi nella pagina Policies (Policy) per tutte le policy gestite. Le politiche AWS gestite includono politiche AWS gestite, politiche gestite delle funzioni lavorative e politiche gestite dai clienti. Puoi visualizzare riepiloghi per queste policy nella pagina Policies (Policy), indipendentemente dal fatto che siano collegate a un utente o a un'altra identità IAM.

Puoi utilizzare le informazioni nei riepiloghi delle policy per comprendere le autorizzazioni che vengono concesse o negate dalla policy. I riepiloghi delle policy facilitano la risoluzione dei problemi e consentono di correggere policy che non forniscono le autorizzazioni previste.

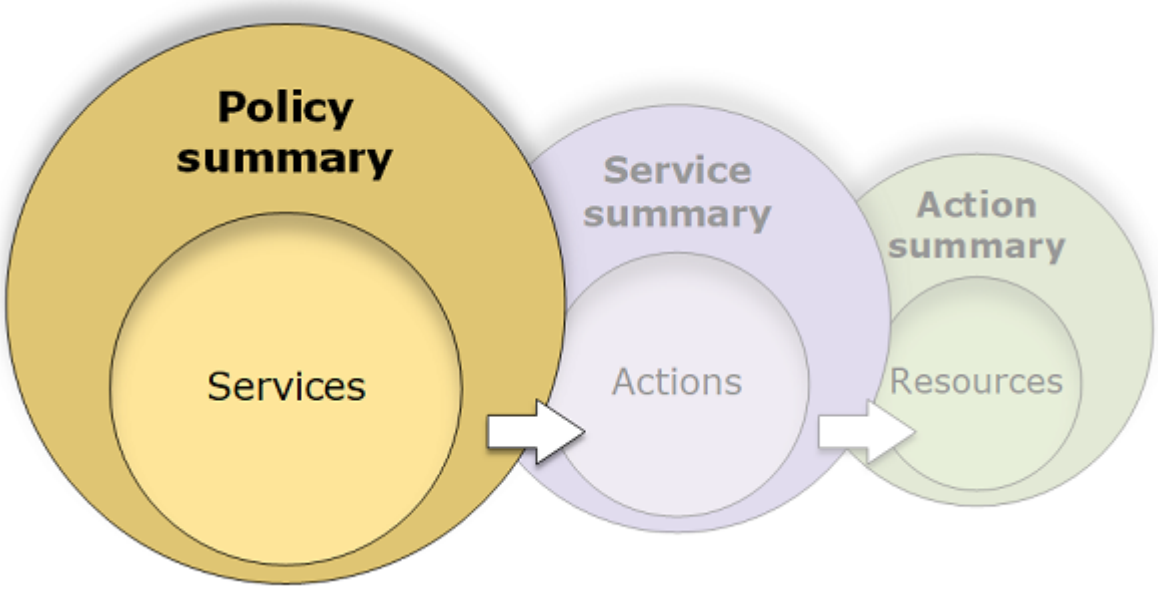
Argomenti

- Riepilogo della policy (elenco di servizi)
- Riepilogo del servizio (elenco di operazioni)

- Riepilogo delle operazioni (elenco di risorse)
- Esempi di riepiloghi di policy

Riepilogo della policy (elenco di servizi)

Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella riepilogo della policy include un elenco di servizi e riepiloghi delle autorizzazioni definite dalla policy scelta.



La tabella di riepilogo della policy è raggruppata in una o più sezioni: Uncategorized services (Servizi non categorizzati), Explicit deny (Rifiuto esplicito) e Allow (Permetti). Se la policy include un servizio che IAM non riconosce, il servizio viene incluso nella sezione Servizi non categorizzati della tabella. Se IAM riconosce il servizio, viene incluso nelle sezioni Rifiuto esplicito o Permetti della tabella, a seconda dell'effetto della policy (`Deny` o `Allow`).

Visualizzazione dei riepiloghi delle policy

È possibile visualizzare i riepiloghi di tutte le policy allegate a un utente scegliendo il nome della policy nella scheda Autorizzazioni nella pagina dei dettagli dell'utente. È possibile visualizzare i riepiloghi di tutte le policy allegate a un ruolo scegliendo il nome della policy nella scheda Autorizzazioni nella pagina dei dettagli dell'utente. È possibile visualizzare il riepilogo della policy per le policy gestite nella pagina Policies (Policy). Se la policy non include un riepilogo, consultare Riepilogo della policy mancante per scoprire perché.

Per visualizzare il riepilogo della policy dalla pagina Policies (Policy)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.

Per visualizzare il riepilogo per una policy collegata a un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Users (Utenti) dal riquadro di navigazione.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, espandere la riga della policy che si desidera visualizzare.

Per visualizzare il riepilogo per una policy collegata a un ruolo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, espandere la riga della policy che si desidera visualizzare.

Modifica delle policy per correggere gli avvisi

Durante la visualizzazione di un riepilogo della policy, è possibile che venga rilevato un errore o che la policy non fornisca le autorizzazioni previste. Non è possibile modificare direttamente un riepilogo

della policy. Tuttavia, è possibile modificare una policy gestita dal cliente utilizzando l'editor visivo della policy, che rileva molti degli stessi errori e avvisi segnalati dal riepilogo della policy. È quindi possibile visualizzare le modifiche nel riepilogo della policy per verificare se sono stati risolti tutti i problemi. Per ulteriori informazioni su come modificare una policy inline, consultare the section called "Modifica delle policy IAM". Non è possibile modificare le politiche AWS gestite.

Per modificare una policy per il riepilogo della policy tramite l'opzione Visivo

1. Aprire il riepilogo della policy come spiegato nelle procedure precedenti.
2. Scegli Modifica.

   Se nella pagina Users (Utenti) si sceglie di modificare una policy gestita dal cliente collegata a tale utente, si viene reindirizzati alla pagina Policies (Policy). È possibile modificare le policy gestite dai clienti solo nella pagina Policies (Policy).
3. Seleziona l'opzione Visivo per visualizzare la rappresentazione visiva modificabile della policy. IAM potrebbe modificare la struttura della policy per ottimizzarla per l'editor visivo e facilitare così la ricerca e la risoluzione di eventuali problemi. Gli avvisi e i messaggi di errore nella pagina possono agevolare la risoluzione di eventuali problemi della policy. Per ulteriori informazioni sul modo in cui IAM modifica la struttura delle policy, consulta Modifica della struttura delle policy.
4. Modifica la policy e seleziona Successivo per visualizzare le modifiche riflesse nel riepilogo della policy. Se sono presenti ancora problemi, selezionare Previous (Precedente) per tornare alla schermata di modifica.
5. Per salvare le modifiche, scegliere Salva modifiche.

Per modificare una policy per il riepilogo della policy con l'opzione JSON

1. Aprire il riepilogo della policy come spiegato nelle procedure precedenti.
2. Utilizza i pulsanti Riepilogo e JSON per confrontare il riepilogo della policy e il documento della policy JSON. È possibile utilizzare queste informazioni per determinare quali righe del documento di policy modificare.
3. Scegli Modifica e quindi seleziona l'opzione JSON per modificare il documento della policy JSON.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'opzione dell'editor Visivo,

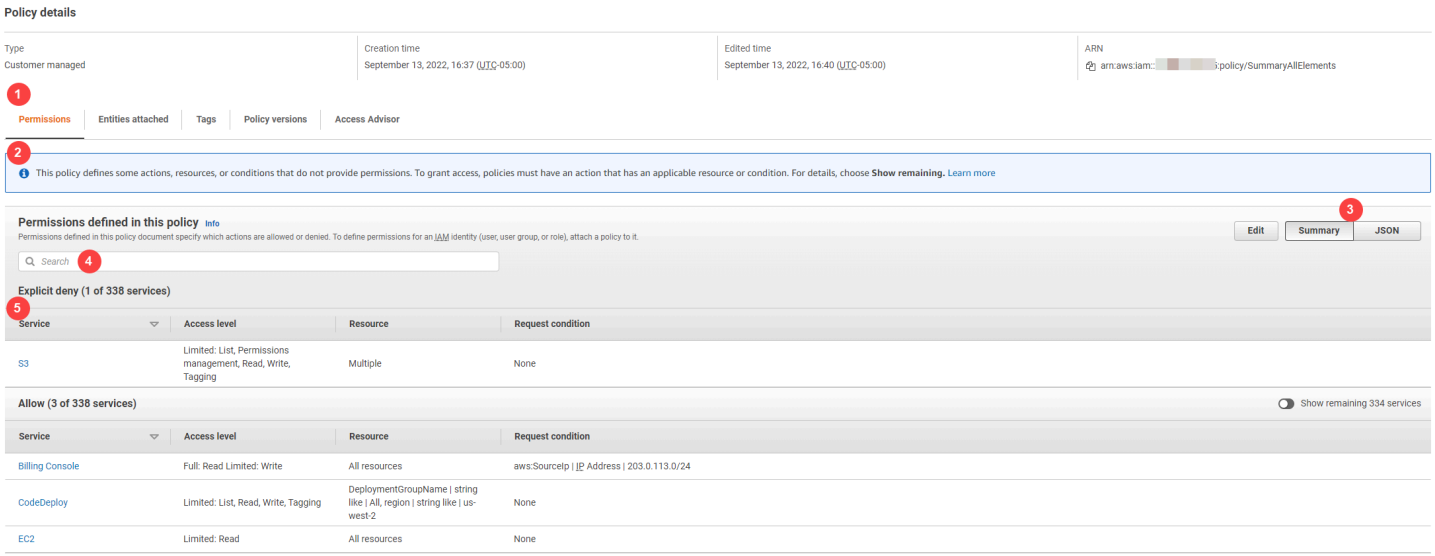
> IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

Se nella pagina Users (Utenti) si sceglie di modificare una policy gestita dal cliente collegata a tale utente, si viene reindirizzati alla pagina Policies (Policy). È possibile modificare le policy gestite dai clienti solo nella pagina Policies (Policy).

4. Modifica la policy. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo). Se sono presenti ancora problemi, selezionare Previous (Precedente) per tornare alla schermata di modifica.
5. Per salvare le modifiche, scegliere Salva modifiche.

Comprendere gli elementi del riepilogo di una policy

Nel seguente esempio di pagina dei dettagli della policy, la SummaryAllElementspolicy è una policy gestita (policy gestita dal cliente) allegata direttamente all'utente. Questa policy è espansa per visualizzare il riepilogo della policy.



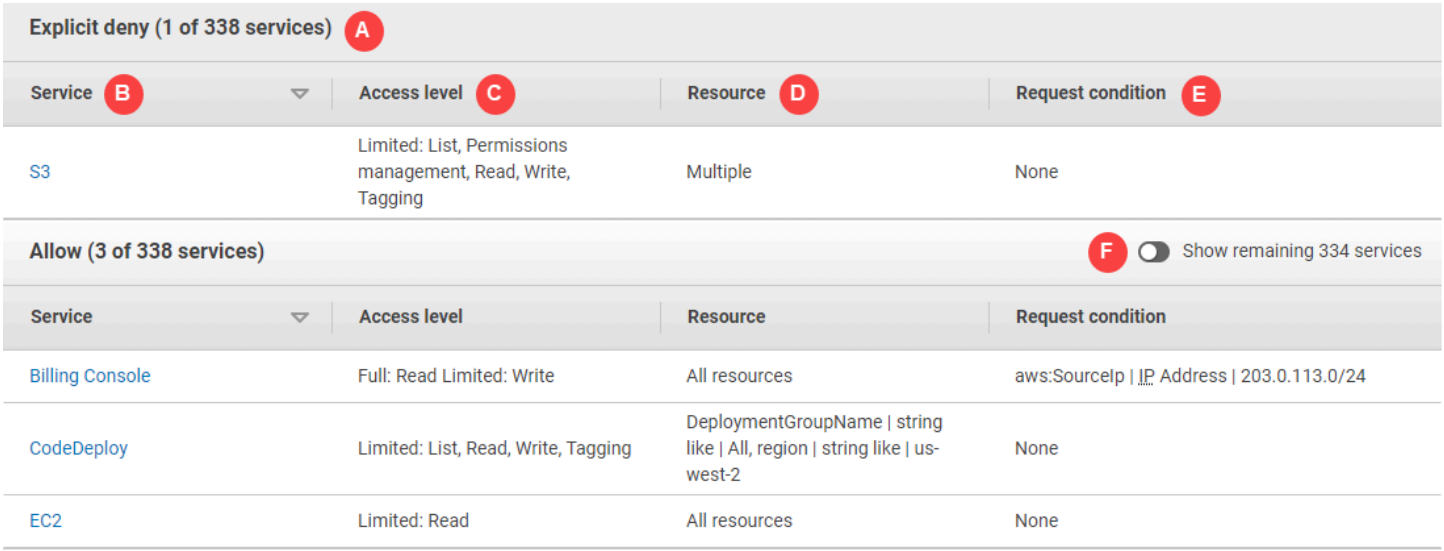
Nell'immagine precedente, il riepilogo della policy è visibile all'interno della pagina Policy:

1. La scheda Autorizzazioni include le autorizzazioni definite nella policy.
2. Se la policy non concede le autorizzazioni a tutte le operazioni, risorse e condizioni definite per il servizio nella policy, viene visualizzato un banner di avviso o errore nella parte superiore della pagina. Il riepilogo della policy include i dettagli sul problema. Per ulteriori informazioni su come

i riepiloghi della policy aiutano a capire e risolvere i problemi delle autorizzazioni che la policy concede, consultare the section called “La policy non concede le autorizzazioni previste”.

3. Utilizza i pulsanti Riepilogo e JSON per passare tra il riepilogo della policy e il documento della policy JSON.
4. Utilizza la casella Cerca per ridurre l'elenco dei servizi e trovare un servizio specifico.
5. La visualizzazione estesa mostra dettagli aggiuntivi della SummaryAllElementspolitica.

La seguente immagine della tabella di riepilogo della politica mostra la SummaryAllElementspolitica estesa nella pagina dei dettagli della politica.



Explicit deny (1 of 338 services) A

Service B	Access level C	Resource D	Request condition E
S3	Limited: List, Permissions management, Read, Write, Tagging	Multiple	None

Allow (3 of 338 services) F Show remaining 334 services

Service	Access level	Resource	Request condition
Billing Console	Full: Read Limited: Write	All resources	aws:SourceIp | IP Address | 203.0.113.0/24
CodeDeploy	Limited: List, Read, Write, Tagging	DeploymentGroupName | string like | All, region | string like | us-west-2	None
EC2	Limited: Read	All resources	None

Nell'immagine precedente, il riepilogo della policy è visibile all'interno della pagina Policy:

A. Per i servizi riconosciuti, IAM li organizza in base al fatto che la policy permetta o rifiuti esplicitamente l'utilizzo del servizio. In questo esempio, la policy include una `Deny` dichiarazione per il servizio Amazon S3 e `Allow` dichiarazioni per i servizi di fatturazione e Amazon EC2. CodeDeploy

B. Servizio: questa colonna riporta i servizi definiti all'interno della policy e fornisce i dettagli per ciascun servizio. Ogni nome di servizio nella tabella di riepilogo della policy è un collegamento alla tabella di riepilogo del servizio illustrata in Riepilogo del servizio (elenco di operazioni). In questo esempio, vengono definite le autorizzazioni per i servizi Amazon S3, Billing CodeDeploy e Amazon EC2.

C. Livello di accesso: questa colonna indica se le operazioni di ciascun livello di accesso (`List`, `Read`, `Write`, `Permission Management` e `Tagging`) dispongono dell'autorizzazione `Full`

o `Limited` definita nella policy. Per ulteriori informazioni ed esempi del riepilogo del livello di accesso, consultare Comprensione dei livelli di accesso nei riepiloghi delle politiche.

- Accesso completo: questa voce indica che il servizio ha accesso a tutte le operazioni entro tutti e quattro i livelli di accesso disponibili per il servizio.
- Se la voce non include Full access (Accesso completo), il servizio ha accesso ad alcune ma non tutte le operazioni per il servizio. L'accesso viene quindi definito dalle seguenti descrizioni per ciascuna delle classificazioni a livello di accesso (`List`, `Read`, `Write`, `Permission Management` e `Tagging`):

  Full (Completo): la policy consente l'accesso a tutte le operazioni all'interno di ciascuna classificazione del livello di accesso elencata. In questo esempio, la policy consente l'accesso a tutte le operazioni `Read` di fatturazione.

  Limited (Limitato): la policy consente l'accesso a una o più operazioni all'interno di ciascuna classificazione del livello di accesso elencata, ma non a tutte. In questo esempio, la policy consente l'accesso ad alcune operazioni `Write` di fatturazione.

D. Risorsa: questa colonna mostra le risorse che la policy specifica per ogni servizio.

- Multiple: la policy include più di una ma non tutte le risorse all'interno del servizio. In questo esempio, l'accesso viene rifiutato esplicitamente a più di una risorsa Amazon S3.
- Tutte le risorse: la policy è definita per tutte le risorse all'interno del servizio. In questo esempio, la policy permette di eseguire le operazioni elencate per tutte le risorse di fatturazione.
- Testo della risorsa: la policy include una risorsa all'interno del servizio. In questo esempio, le azioni elencate sono consentite solo sulla risorsa. `DeploymentGroupName` CodeDeploy A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN o il tipo di risorsa definita.

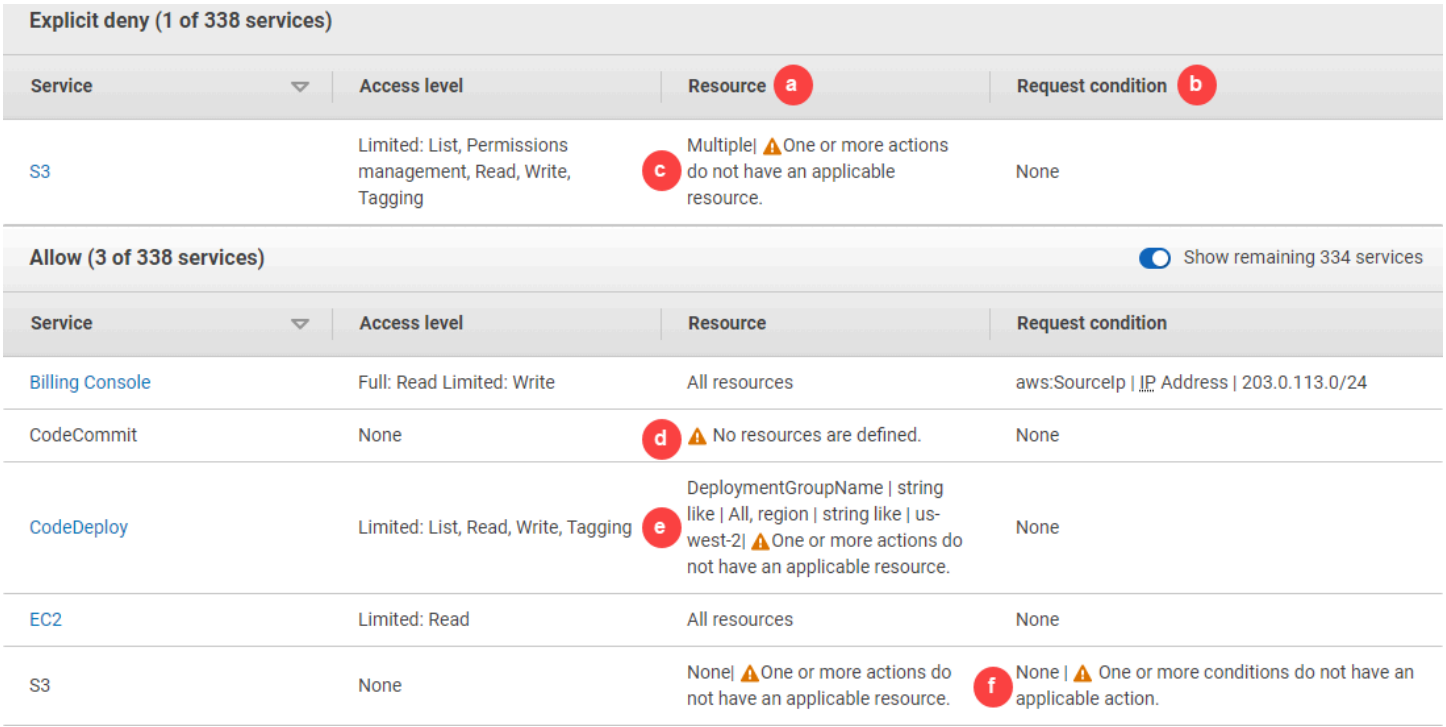
> **Note**
>
> Questa colonna può includere una risorsa da un altro servizio. Se l'istruzione di policy che include la risorsa non include entrambe le operazioni e risorse dallo stesso servizio, la policy include le risorse non corrispondenti. IAM non visualizza avvisi per le risorse non corrispondenti al momento della creazione di una policy o della visualizzazione di una policy nel riepilogo della policy. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per verificare meglio le policy, eseguire sempre un test tramite il simulatore di policy.

E. Condizioni richiesta: questa colonna indica se i servizi o le operazioni associati alla risorsa sono soggetti a condizioni.

- Nessuna: la policy non include condizioni per il servizio. In questo esempio non vengono applicate condizioni alle operazioni rifiutate nel servizio Amazon S3.
- Testo della condizione: la policy include una condizione per il servizio. In questo esempio, le operazioni di Fatturazione elencate sono consentite solo se l'indirizzo IP di origine corrisponde a `203.0.113.0/24`.
- Multiple: la policy include più di una condizione per il servizio. Per visualizzare tutte le condizioni multiple per la policy, seleziona JSON per visualizzare il documento della policy.

F. Mostra servizi rimanenti: attiva/disattiva questo pulsante per espandere la tabella e includere i servizi che non sono definiti dalla policy. Questi servizi vengono rifiutati implicitamente (o rifiutati per impostazione predefinita) all'interno della policy. Tuttavia, un'istruzione in un'altra policy potrebbe permettere o rifiutare esplicitamente l'utilizzo del servizio. Il riepilogo della policy fornisce un elenco delle autorizzazioni di una singola policy. Per informazioni su come il AWS servizio decide se consentire o rifiutare una determinata richiesta, consultaLogica di valutazione delle policy.

Quando una policy o un elemento all'interno della policy non concede autorizzazioni, IAM vengono forniti ulteriori avvisi e informazioni nel riepilogo della policy. La seguente tabella di riepilogo delle politiche mostra la versione estesa dei servizi Mostra i servizi rimanenti nella pagina dei dettagli della SummaryAllElementspolitica con i possibili avvisi.



Explicit deny (1 of 338 services)

Service	Access level	Resource a	Request condition b
S3	Limited: List, Permissions management, Read, Write, Tagging	c Multiple| ⚠ One or more actions do not have an applicable resource.	None

Allow (3 of 338 services) — Show remaining 334 services

Service	Access level	Resource	Request condition
Billing Console	Full: Read Limited: Write	All resources	aws:SourceIp | IP Address | 203.0.113.0/24
CodeCommit	None	d ⚠ No resources are defined.	None
CodeDeploy	Limited: List, Read, Write, Tagging	e DeploymentGroupName | string like | All, region | string like | us-west-2| ⚠ One or more actions do not have an applicable resource.	None
EC2	Limited: Read	All resources	None
S3	None	None| ⚠ One or more actions do not have an applicable resource.	f None | ⚠ One or more conditions do not have an applicable action.

Nell'immagine precedente, è possibile visualizzare tutti i servizi che includono operazioni, risorse o condizioni definite senza autorizzazioni.

a. Avvisi risorse: per i servizi che non forniscono autorizzazioni per tutte le operazioni o risorse incluse, viene visualizzato uno dei seguenti avvisi nella colonna Risorsa della tabella:

   - ⚠ No resources are defined (Nessuna risorsa definita) : indica che il servizio ha definito le operazioni, ma nessuna risorsa supportata è inclusa nella policy.
   - ⚠ One or more actions do not have an applicable resource (Una o più operazioni non hanno una risorsa applicabile) : indica che il servizio ha definito le operazioni, ma che alcune di esse non hanno una risorsa supportata.
   - ⚠ One or more resources do not have an applicable action (Una o più risorse non hanno un'operazione applicabile) : indica che il servizio ha definito le risorse, ma che alcune di esse non hanno un'operazione di supporto.

   Se un servizio include sia operazioni senza una risorsa applicabile sia risorse con una risorsa applicabile, viene visualizzato solo l'avviso Una o più risorse non hanno un'operazione applicabile.

Questo perché quando si visualizza il riepilogo del servizio per il servizio, le risorse che non si applicano a nessuna operazione non vengono visualizzate. Per l'operazione `ListAllMyBuckets`, questa policy include l'ultimo avvertimento perché l'operazione non supporta le autorizzazioni a livello di risorsa e non supporta la chiave di condizione `s3:x-amz-acl`. Se si risolve il problema della risorsa o della condizione, il problema rimanente appare in un avviso dettagliato.

b. Avvisi di condizione richiesta: per i servizi che non forniscono autorizzazioni per tutte le condizioni incluse, viene visualizzato uno dei seguenti avvisi nella colonna Condizione richiesta della tabella:

   - 

     One or more actions do not have an applicable condition (Una o più operazioni non hanno una condizione applicabile) : indica che il servizio ha definito le operazioni, ma che alcune di esse non hanno una condizione supportata.

   - 

     One or more conditions do not have an applicable action (Una o più condizioni non hanno un'operazione applicabile) : indica che il servizio ha definito le condizioni, ma che alcune di esse non hanno un'operazione di supporto.

c. Multiple |

   One or more actions do not have an applicable resource (Multiple | Una o più operazioni non hanno una risorsa applicabile). : l'istruzione `Deny` per Amazon S3 include più di una risorsa. Include anche più di un'operazione e alcune operazioni supportano le risorse, altre no. Per visualizzare questa policy, consulta the section called “Documento di policy JSON SummaryAllElements”. In questo caso, la policy include tutte le operazioni Amazon S3 e vengono rifiutate solo le operazioni che possono essere eseguite per un oggetto bucket o un bucket.

d. 

   Nessuna risorsa definita: il servizio ha definito le operazioni, ma nella policy non sono incluse risorse supportate e pertanto il servizio non fornisce autorizzazioni. In questo caso, la politica include CodeCommit azioni ma non CodeCommit risorse.

e. DeploymentGroupName | string like | All, region | string like | us-west-2

   | Una o più azioni non hanno una risorsa applicabile. : il servizio dispone di una operazione definita e di almeno un'altra operazione senza una risorsa di supporto.

f. Nessuna |

Una o più condizioni non hanno un'operazione applicabile. : il servizio dispone almeno di una chiave di condizione che non ha un'operazione di supporto.

Documento di policy JSON SummaryAllElements

La SummaryAllElementspolicy non è pensata per essere utilizzata dall'utente per definire le autorizzazioni nel proprio account. Al contrario, è inclusa per dimostrare gli errori e gli avvisi che possono verificarsi durante la visualizzazione di una policy di riepilogo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "billing:Get*",
                "payments:List*",
                "payments:Update*",
                "account:Get*",
                "account:List*",
                "cur:GetUsage*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "203.0.113.0/24"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::customer",
                "arn:aws:s3:::customer/*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshots"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codedploy:*",
                "codecommit:*"
            ],
            "Resource": [
                "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*",
                "arn:aws:codebuild:us-east-1:123456789012:project/my-demo-project"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetObject",
                "s3:DeletObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::developer_bucket",
                "arn:aws:s3:::developer_bucket/*",
                "arn:aws:autoscling:us-east-2:123456789012:autoscalgrp"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ],
                    "s3:prefix": [
                        "custom",
                        "other"
                    ]
                }
            }
```

```
        }
    ]
}
```

Comprensione dei livelli di accesso nei riepiloghi delle politiche

AWS riepilogo dei livelli di accesso

I riepiloghi delle policy includono un riepilogo del livello di accesso che descrive le autorizzazioni operative definite per ciascun servizio menzionato nella policy. Per informazioni sui riepiloghi delle policy, consultare Informazioni sulle autorizzazioni concesse da una policy. I riepiloghi dei livelli di accesso indicano se per le operazioni di ciascun livello di accesso (`List`, `Read`, `Tagging`, `Write`, and `Permissions management`) sono definite autorizzazioni `Full` o `Limited` nella policy. Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, vedere Azioni, risorse e chiavi di condizione per AWS i servizi.

L'esempio seguente descrive l'accesso fornito da una policy per i servizi in questione. Per esempi di documenti completi della policy JSON e i relativi riepiloghi, consultare Esempi di riepiloghi di policy.

Servizio	Livello di accesso	Questa policy fornisce quanto segue
IAM	Accesso completo ad	Accedi a tutte le operazioni all'interno del servizio IAM.
CloudWatch	Completo: elenco	Accesso a tutte CloudWatch le azioni nel livello di `List` accesso, ma nessun accesso alle azioni con classificazione a livello di `Permissions management` accesso o. `Read` `Write`
Data Pipeline	Limitato: elenco, lettura	Accesso ad almeno una ma non a tutte AWS Data Pipeline le azioni nel livello di `Read` accesso `List` e, ma non alle `Permissions management` azioni `Write` o.
EC2	Completo: elenco, lettura Limitato: scrittura	Accesso a tutte le operazioni `List` e `Read` di Amazon EC2 e accesso ad almeno una, ma non a tutte le operazioni `Write` di Amazon EC2, ma nessun accesso alle operazioni con la

Servizio	Livello di accesso	Questa policy fornisce quanto segue
		classificazione a livello di accesso `Permissions management` .
S3	Limitato: lettura, scrittura, gestione autorizzazioni	Accesso ad almeno una, ma non a tutte le operazioni Amazon S3, `Read`, `Write` e `Permissions management` .
CodeDeploy	(vuoto)	Accesso sconosciuto, perché IAM non riconosce questo servizio.
API Gateway	Nessuno	Nessun accesso è definito nella policy.
CodeBuild	⚠ Non viene definita nessuna operazione.	Nessun accesso, perché non sono definite operazioni per il servizio. Per ulteriori informazioni su questo problema e sulla sua risoluzione, consultare the section called “La policy non concede le autorizzazioni previste”.

Come spiegato in precedenza,, Full access (Accesso completo) indica che la policy concede l'accesso a tutte le operazioni all'interno del servizio. Le policy che forniscono accesso ad alcune ma non a tutte le operazioni all'interno di un servizio sono ulteriormente raggruppate in base alla classificazione del livello di accesso. Tale differenza viene indicata da uno dei seguenti raggruppamenti a livello di accesso:

- Full (Completo): la policy consente l'accesso a tutte le operazioni all'interno della classificazione specificata per il livello di accesso.
- Limited (Limitato): la policy consente l'accesso a una o più operazioni all'interno della classificazione specificata per il livello di accesso, ma non a tutte.
- None (Nessuno): la policy non fornisce alcun accesso.
- (vuoto): IAM non riconosce questo servizio. Se il nome del servizio include un errore ortografico, la policy non concederà l'accesso al servizio. Se il nome è corretto, il servizio potrebbe non supportare i riepiloghi della policy o potrebbe essere in anteprima. In questo caso, la policy potrebbe fornire l'accesso, ma questo non può essere visualizzato nel riepilogo della policy. Per

richiedere il riepilogo della policy per un servizio disponibile a livello generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.

I riepiloghi dei livelli di accesso che includono un accesso limitato (parziale) alle azioni sono raggruppati utilizzando le classificazioni dei livelli di AWS accesso`List`,, `ReadTagging`, `Write` o. `Permissions management`

AWS livelli di accesso

AWS definisce le seguenti classificazioni dei livelli di accesso per le azioni in un servizio:

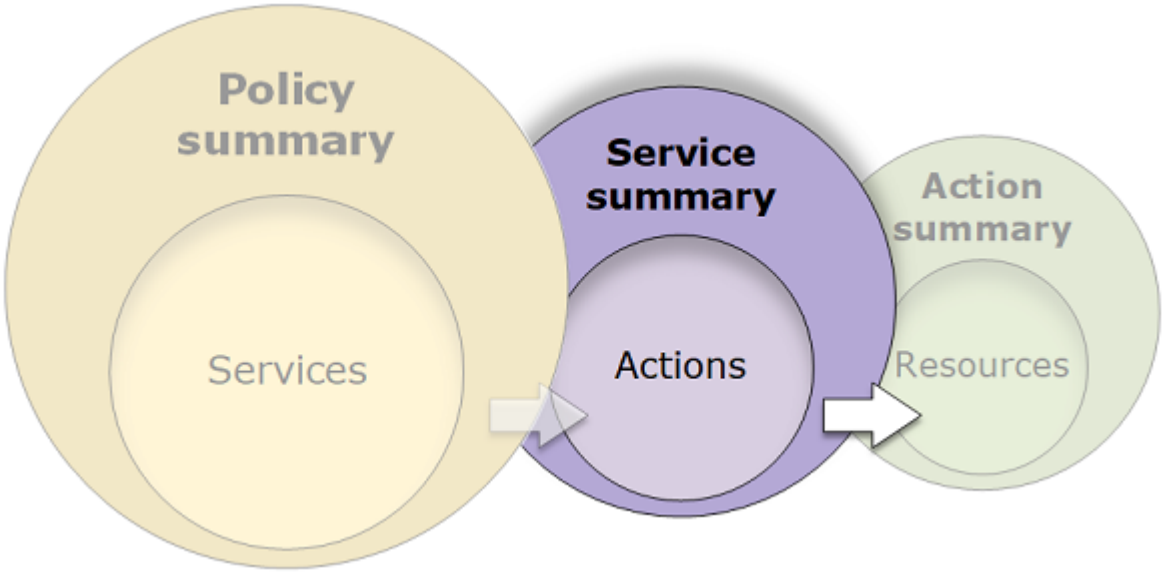
- List (Elenco): autorizzazione a elencare le risorse all'interno del servizio per determinare l'esistenza di un oggetto. Le operazioni con questo livello di accesso possono elencare gli oggetti, ma consentono di visualizzare i contenuti di una risorsa. Ad esempio, l'operazione Amazon S3 `ListBucket` ha un livello di accesso di tipo Elenco.
- Read (Lettura): autorizzazione a leggere ma non a modificare i contenuti e gli attributi delle risorse del servizio. Ad esempio, le operazioni di Amazon S3 `GetObject` e `GetBucketLocation` hanno un livello di accesso Lettura.
- Tagging: autorizzazione per eseguire operazioni che modificano solo lo stato di tag delle risorse. Ad esempio, le operazioni IAM `TagRole` e `UntagRole` dispongono del livello di accesso Tagging, in quanto consentono solo l'aggiunta e la rimozione di tag da un ruolo. Tuttavia, l'operazione `CreateRole` consente il tagging di una risorsa del ruolo al momento della creazione di tale ruolo. Poiché l'operazione non aggiunge solo un tag, dispone del livello di accesso `Write`.
- Write (Scrittura): autorizzazione a creare, eliminare o modificare le risorse del servizio. Ad esempio, le operazioni Amazon S3 `CreateBucket`, `DeleteBucket` e `PutObject` hanno il livello di accesso Scrittura. Le operazioni `Write` potrebbero anche consentire la modifica di un tag della risorsa. Tuttavia, un'operazione che consente solo modifiche ai tag dispone del livello di accesso `Tagging`.
- Permissions management (Gestione autorizzazioni): autorizzazione a concedere o modificare le autorizzazioni a livello di risorsa nel servizio. Ad esempio, la maggior parte AWS Organizations delle azioni IAM e di quelle come Amazon S3 `DeleteBucketPolicy` hanno il livello di accesso alla gestione delle autorizzazioni. `PutBucketPolicy`

> **Suggerimento**
>
> Per migliorare la sicurezza delle tue politiche Account AWS, limita o monitora regolarmente le politiche che includono la classificazione dei livelli di accesso alla gestione delle autorizzazioni.

Per visualizzare la classificazione del livello di accesso per tutte le azioni di un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Riepilogo del servizio (elenco di operazioni)

Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella riepilogo del servizio include un elenco di operazioni e riepiloghi delle autorizzazioni definite dalla policy per il servizio selezionato.



È possibile visualizzare un riepilogo di servizio per ogni servizio elencato nel riepilogo della policy che concede autorizzazioni. La tabella è raggruppata in Uncategorized actions (Operazioni non categorizzate), Uncategorized resource types (Tipi di risorse non categorizzate) e sezioni di livello di accesso. Se la policy include un'operazione che IAM non riconosce, l'operazione sarà inclusa nella sezione Operazioni non categorizzate della tabella. Se IAM riconosce l'operazione, è inclusa in una delle sezioni della tabella dei livelli di accesso (Elenca, Lettura, Scrittura e Gestione autorizzazioni). Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, vedere Azioni, risorse e chiavi di condizione per AWS i servizi.

Visualizzazione dei riepiloghi del servizio

È possibile visualizzare il riepilogo dei servizi per le policy gestite nella pagina Policy.

Per visualizzare il riepilogo del servizio per una policy gestita

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.
5. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

Per visualizzare il riepilogo del servizio per una policy collegata a un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Utenti.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Utenti si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Scegli Riepilogo. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

> **Note**
>
> Se la policy selezionata è una policy inline collegata direttamente all'utente, appare la tabella di riepilogo del servizio. Se la policy è una policy inline collegata da un gruppo,

si passa al documento della policy JSON per quel gruppo. Se la policy è una policy gestita, si viene reindirizzati al riepilogo del servizio per quella policy nella pagina Policies (Policy).

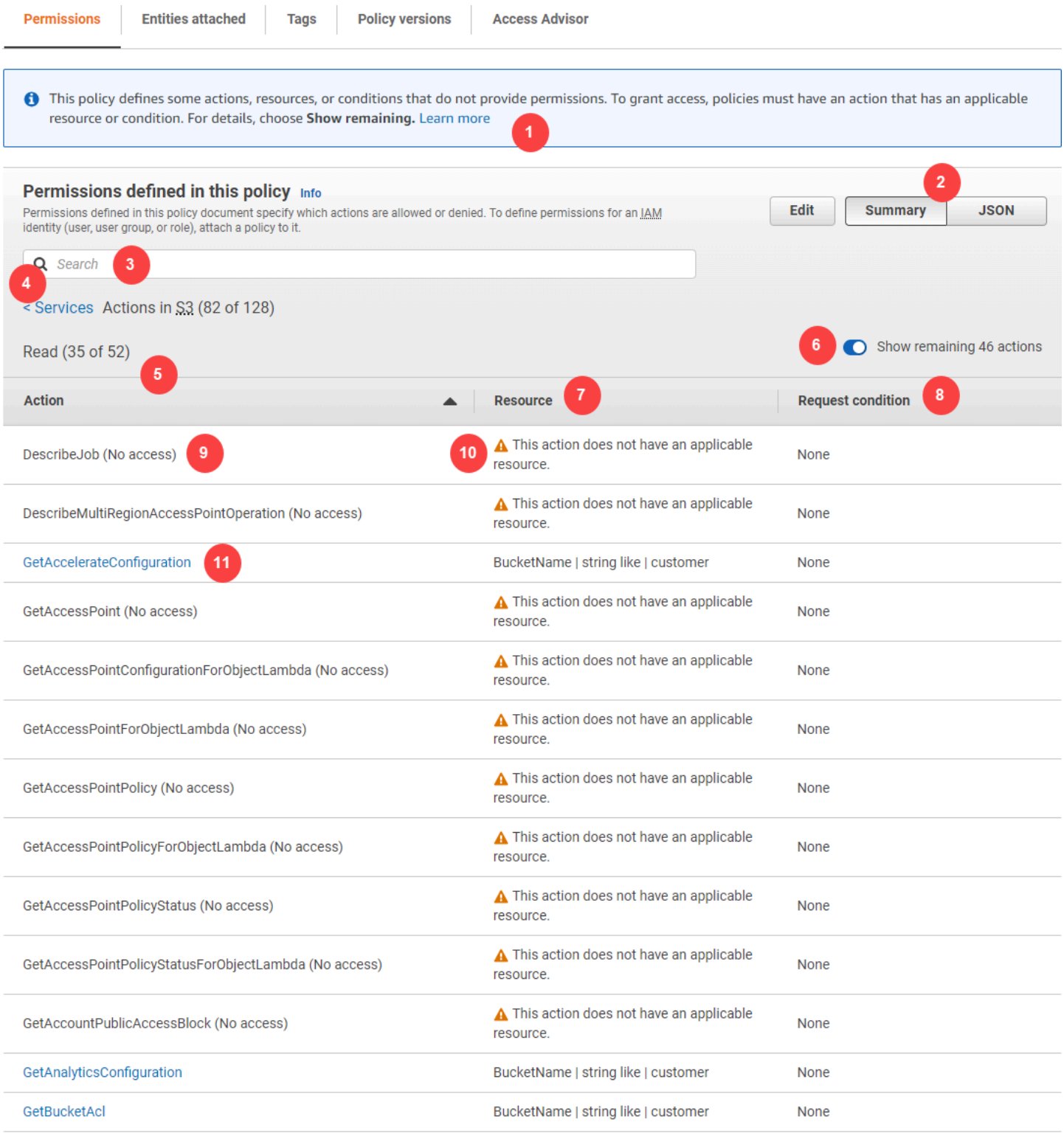
Per visualizzare il riepilogo del servizio per una policy collegata a un ruolo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Roles (Ruoli) dal riquadro di navigazione.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Ruoli si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

Informazioni sugli elementi del riepilogo di un servizio

L'esempio seguente è il riepilogo del servizio per le operazioni Amazon S3 consentite dal riepilogo della policy. Le operazioni per questo servizio sono raggruppate per livello di accesso. Ad esempio, sono definite 35 operazioni di lettura rispetto alle 52 operazioni di lettura totali disponibili per il servizio.



La pagina del riepilogo del servizio per una policy gestita include le seguenti informazioni:

1. Se la policy non concede le autorizzazioni a tutte le operazioni, risorse e condizioni definite per il servizio nella policy, quindi un banner di avviso viene visualizzato nella parte superiore della

pagina. Il riepilogo del servizio include i dettagli sul problema. Per ulteriori informazioni su come i riepiloghi della policy aiutano a capire e risolvere i problemi delle autorizzazioni che la policy concede, consultare the section called “La policy non concede le autorizzazioni previste”.

2. Seleziona JSON per visualizzare ulteriori dettagli sulla policy. È possibile eseguire questa operazione per visualizzare tutte le condizioni applicate alle operazioni. (Se si sta visualizzando il riepilogo del servizio per una policy inline collegata direttamente a un utente, è necessario chiudere la finestra del dialogo del riepilogo del servizio e tornare al riepilogo della policy per accedere al documento della policy JSON.)
3. Per visualizzare il riepilogo per una risorsa operazione, digita le parole chiave nella casella Cerca per ridurre l'elenco delle operazioni disponibili.
4. Accanto alla freccia Servizi viene visualizzato il nome del servizio (in questo caso S3). Il riepilogo del servizio per questo servizio include l'elenco delle operazioni consentite o negate definite nella policy. Se il servizio viene visualizzato in (Negazione esplicita) nella scheda Autorizzazioni, le operazioni elencate nella tabella di riepilogo del servizio vengono negate esplicitamente. Se il servizio viene visualizzato in Consenti nella scheda Autorizzazioni, le operazioni elencate nella tabella di riepilogo del servizio vengono consentite.
5. Operazione: questa colonna elenca le operazioni definite nella policy e fornisce le risorse e le condizioni per ciascuna operazione. Se la policy concede o nega le autorizzazioni all'operazione, il nome dell'operazione si collega alla tabella riepilogo dell'operazione. La tabella raggruppa queste operazioni in almeno una o fino a cinque sezioni, a seconda del livello di accesso che la policy consente o nega. Le sezioni sono Elenco, Lettura, Scrittura, Gestione autorizzazionie Tagging. Il conteggio indica il numero di operazioni riconosciute che forniscono le autorizzazioni per ogni livello di accesso. Il totale è il numero di operazioni note per il servizio. In questo esempio, 35 operazioni forniscono le autorizzazioni su un totale di 52 operazioni di lettura Amazon S3 note. Per visualizzare la classificazione del livello di accesso assegnata a ciascuna azione in un servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.
6. Mostra operazioni rimanenti: attiva/disattiva questo pulsante per espandere o nascondere la tabella e includere le operazioni che sono note ma non forniscono le autorizzazioni per questo servizio. L'attivazione o la disattivazione del pulsante visualizza inoltre avvisi per gli elementi che non forniscono le autorizzazioni.
7. Risorsa: questa colonna mostra le risorse che la policy definisce per il servizio. IAM non verifica se la risorsa si applica a ciascuna operazione. In questo esempio, le operazioni nel servizio Amazon S3 sono consentite solo nella risorsa del bucket Amazon S3 `developer_bucket`. A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN come

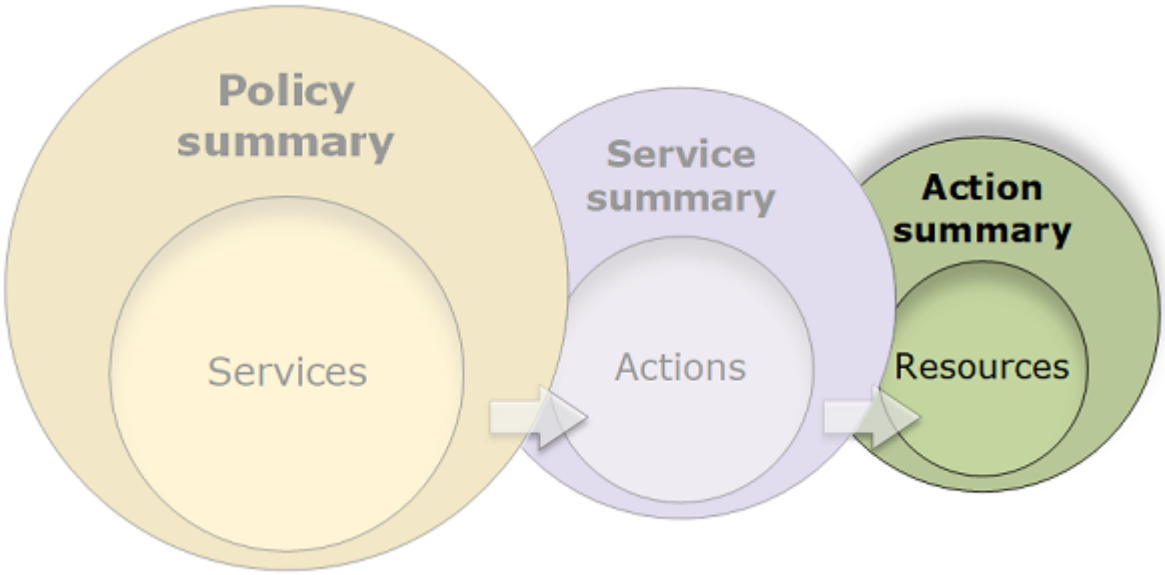
`arn:aws:s3:::developer_bucket/*`, oppure il tipo di risorsa definita, come `BucketName = developer_bucket`.

> **Note**
>
> Questa colonna può includere una risorsa da un altro servizio. Se l'istruzione di policy che include la risorsa non include entrambe le operazioni e risorse dallo stesso servizio, la policy include le risorse non corrispondenti. IAM non avvisa riguardo alle risorse non corrispondenti al momento della creazione di una policy oppure quando si visualizza una policy nel riepilogo del servizio. IAM inoltre non indica se l'operazione è valida per le risorse, solo se il servizio corrisponde. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per verificare meglio le policy, eseguire sempre un test tramite il simulatore di policy.

8. Condizioni richiesta: questa colonna mostra se le operazioni che sono associate alla risorsa sono soggette a condizioni. Per ulteriori informazioni su queste condizioni, seleziona JSON per visualizzare il documento della policy JSON.
9. Nessun accesso: questa policy include un'operazione che non fornisce autorizzazioni.
10. Avviso risorsa: per operazioni con risorse che non forniscono autorizzazioni complete, viene visualizzato uno dei seguenti avvisi:
    - Questa operazione non supporta le autorizzazioni a livello di risorsa. Richiede un carattere jolly (*) per la risorsa. : indica che la policy include le autorizzazioni a livello di risorsa, ma deve includere `"Resource": ["*"]` per fornire le autorizzazioni per questa operazione.
    - This action does not have an applicable resource (Questa operazione non ha una risorsa applicabile) : indica che l'operazione è inclusa nella policy senza una risorsa supportata.
    - This action does not have an applicable resource and condition (Questa operazione non ha una risorsa e una condizione applicabili) : indica che l'operazione è inclusa nella policy senza una risorsa supportata e senza una condizione supportata. In questo caso, sussiste anche una condizione inclusa nella policy per questo servizio, ma non ci sono condizioni che si applicano a questa operazione.
11. Le operazioni che forniscono le autorizzazioni includono un collegamento al riepilogo dell'operazione.

Riepilogo delle operazioni (elenco di risorse)

Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella del riepilogo dell'operazione include un elenco di risorse e le condizioni associate che si applicano a un'operazione prescelta.



Per visualizzare un riepilogo dell'operazione per ciascuna operazione che consente le autorizzazioni, selezionare il collegamento nel riepilogo dei servizi. La tabella di riepilogo dell'operazione include dettagli sulla risorsa, compresi la Region (Regione) e l'Account. È possibile anche visualizzare le condizioni applicabili a ogni risorsa. Questo illustra le condizioni che si applicano ad alcune risorse ma non altre.

Visualizzazione dei riepiloghi delle operazioni

È possibile visualizzare il riepilogo delle operazioni per le policy gestite, qualsiasi policy associata a un utente e qualsiasi policy associata a un ruolo nella pagina Policy.

Per visualizzare il riepilogo delle operazioni per una policy gestita

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Nella pagina Dettagli della policy per la policy, visualizza la scheda Autorizzazioni per consultare il riepilogo della policy.

5. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.
6. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

Per visualizzare il riepilogo dell'operazione per una policy collegata a un utente

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Selezionare Users (Utenti) dal riquadro di navigazione.
3. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per l'utente, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate all'utente direttamente o da un gruppo.
5. Nella tabella di policy per l'utente, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Utenti si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.

   > **Note**
   >
   > Se la policy selezionata è una policy inline collegata direttamente all'utente, appare la tabella di riepilogo del servizio. Se la policy è una policy inline collegata da un gruppo, si passa al documento della policy JSON per quel gruppo. Se la policy è una policy gestita, si viene reindirizzati al riepilogo del servizio per quella policy nella pagina Policies (Policy).
7. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

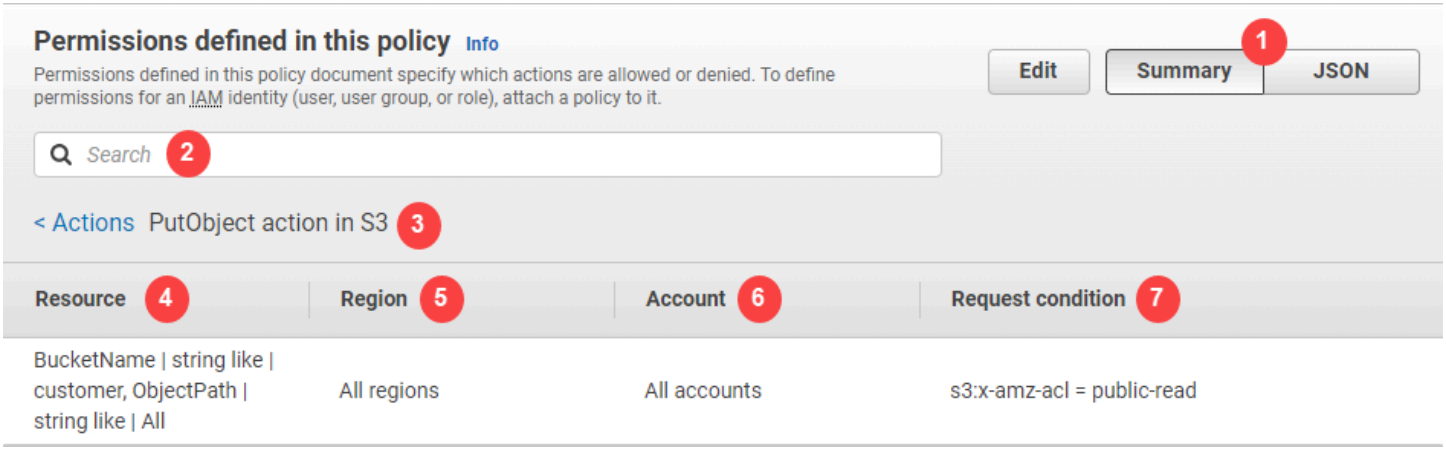
Per visualizzare il riepilogo dell'operazione per una policy collegata a un ruolo

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Nell'elenco di ruoli, selezionare il nome del ruolo la cui policy si desidera visualizzare.
4. Nella pagina Summary (Riepilogo) per il ruolo, visualizzare la scheda Permissions (Autorizzazioni) per visualizzare l'elenco di policy collegate al ruolo.
5. Nella tabella di policy per il ruolo, scegli il nome della policy che si desidera visualizzare.

   Se nella pagina Ruoli si sceglie di visualizzare il riepilogo dei servizi per una policy collegata a tale utente, si viene reindirizzati alla pagina Policy. È possibile visualizzare i riepiloghi dei servizi solo nella pagina Policy.
6. Nell'elenco dei servizi del riepilogo della policy, selezionare il nome del servizio che si desidera modificare.
7. Nell'elenco delle operazioni del riepilogo del servizio, selezionare il nome dell'operazione che si desidera visualizzare.

Informazioni sugli elementi del riepilogo di un'operazione

L'esempio seguente è il riepilogo dell'operazione (di scrittura) `PutObject` dal riepilogo del servizio Amazon S3 (consulta Riepilogo del servizio (elenco di operazioni)). Per questa operazione, la policy definisce più condizioni su una singola risorsa.



La pagina di riepilogo dell'operazione include le informazioni riportate di seguito:

1. Seleziona JSON per visualizzare ulteriori dettagli sulla policy, ad esempio le condizioni multiple applicate alle operazioni. Se stai visualizzando il riepilogo del servizio per una policy in linea collegata direttamente a un utente, la procedura potrebbe differire. Per accedere al documento di policy JSON in questo caso, è necessario chiudere la finestra del dialogo di riepilogo delle operazioni e tornare al riepilogo della policy.)
2. Per visualizzare il riepilogo per una risorsa specifica, digita le parole chiave nella casella Cerca per ridurre l'elenco delle risorse disponibili.
3. Accanto alla freccia indietro delle azioni viene visualizzato il nome del servizio e dell'azione nel formato `action name action in service` (in questo caso PutObjectl'azione in S3). Il riepilogo dell'operazione per questo servizio include l'elenco delle risorse definite nella policy.
4. Risorsa: questa colonna elenca le risorse che la policy definisce per il servizio scelto. In questo esempio, l'PutObjectazione è consentita su tutti i percorsi degli oggetti, ma solo sulla risorsa bucket `developer_bucket` Amazon S3. A seconda delle informazioni che il servizio fornisce a IAM, è possibile che venga visualizzato un ARN come `arn:aws:s3:::developer_bucket/*`, oppure il tipo di risorsa definita, come `BucketName = developer_bucket, ObjectPath = All`.
5. Regione: questa colonna mostra la regione in cui la risorsa viene definita. Le risorse possono essere definite per tutte le regioni o per una singola regione. Non possono esistere in più di una regione specifica.
   - Tutte le regioni: le operazioni associate alla risorsa si applicano a tutte le regioni. In questo esempio, l'operazione appartiene a un servizio globale, Amazon S3. Le operazioni che appartengono a servizi globali si applicano a tutte le regioni.
   - Testo regione: le operazioni associate alla risorsa si applicano a una regione. Ad esempio, una policy può specificare la regione `us-east-2` per una risorsa.
6. Account: questa colonna indica se i servizi o le operazioni associati alla risorsa si applicano a un determinato account. Le risorse possono esistere in tutti gli account o in un singolo account. Non possono esistere in più di un determinato account.
   - Tutti gli account: le operazioni associate alla risorsa si applicano a tutti gli account. In questo esempio, l'operazione appartiene a un servizio globale, Amazon S3. Le operazioni che appartengono a servizi globali si applicano a tutti gli account.
   - Questo account: le operazioni associate alla risorsa si applicano solo all'account corrente.
   - Numero di account: le operazioni associate alla risorsa si applicano a un account (uno al quale non è stato effettuato l'accesso). Ad esempio, se una policy specifica l'account`123456789012` per una risorsa, quindi il numero di account viene visualizzato nel riepilogo della policy.

7. Condizione richiesta: questa colonna mostra se le operazioni associate alla risorsa sono soggette a condizioni. Questo esempio include la condizione `s3:x-amz-acl = public-read`. Per ulteriori informazioni su queste condizioni, seleziona JSON per visualizzare il documento della policy JSON.

Esempi di riepiloghi di policy

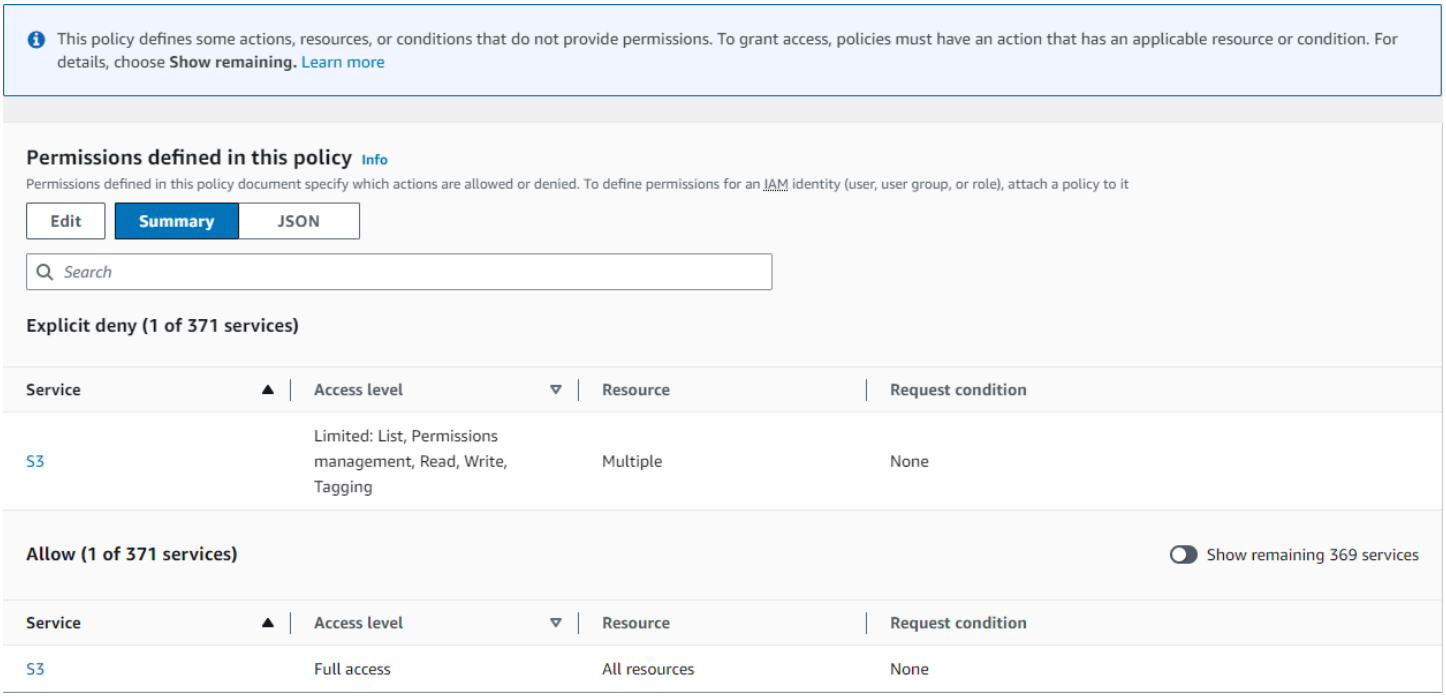
Gli esempi seguenti includono le policy JSON con i relativi riepiloghi di policy, i riepiloghi dei servizi e i riepiloghi delle operazioni, per aiutarti a comprendere le autorizzazioni concesse tramite una policy.

Politica 1: DenyCustomerBucket

Questa policy illustra un permesso e un rifiuto per lo stesso servizio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": ["s3:*"],
            "Resource": ["*"]
        },
        {
            "Sid": "DenyCustomerBucket",
            "Action": ["s3:*"],
            "Effect": "Deny",
            "Resource": ["arn:aws:s3:::customer", "arn:aws:s3:::customer/*" ]
        }
    ]
}
```

DenyCustomerBucketRiepilogo della politica:



DenyCustomerBucket Riepilogo del servizio S3 (negazione esplicita):

< Services Actions in S3 (82 of 130) Show remaining 48 actions

Read (35 of 53)

Action	Resource	Request condition
GetAccelerateConfiguration	BucketName| string like |customer	None
GetAnalyticsConfiguration	BucketName| string like |customer	None
GetBucketAcl	BucketName| string like |customer	None
GetBucketCORS	BucketName| string like |customer	None
GetBucketLocation	BucketName| string like |customer	None
GetBucketLogging	BucketName| string like |customer	None
GetBucketNotification	BucketName| string like |customer	None
GetBucketObjectLockConfiguration	BucketName| string like |customer	None
GetBucketOwnershipControls	BucketName| string like |customer	None
GetBucketPolicy	BucketName| string like |customer	None
GetBucketPolicyStatus	BucketName| string like |customer	None
GetBucketPublicAccessBlock	BucketName| string like |customer	None
GetBucketRequestPayment	BucketName| string like |customer	None
GetBucketTagging	BucketName| string like |customer	None
GetBucketVersioning	BucketName| string like |customer	None
GetBucketWebsite	BucketName| string like |customer	None

GetObject (Leggi) Riepilogo delle azioni:

< Actions GetObject action in S3

Resource	Region	Account	Request condition
BucketName| string like |customer, ObjectPath| string like |All	-	All accounts	None

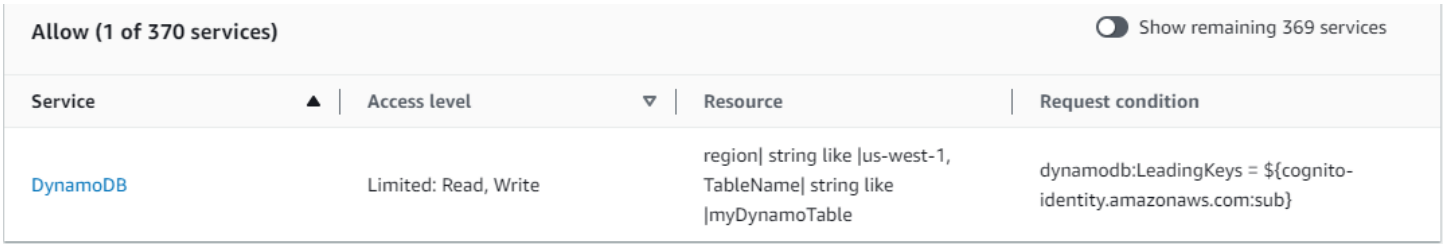
Politica 2: DynamoDbRowCognito ID

Questa policy consente l'accesso a livello di riga ad Amazon DynamoDB in base all'ID Amazon Cognito dell'utente.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-west-1:123456789012:table/myDynamoTable"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": [
                        "${cognito-identity.amazonaws.com:sub}"
                    ]
                }
            }
        }
    ]
}
```

DynamoDbRowCognitoRiepilogo della politica ID:



DynamoDbRowCognitoRiepilogo del servizio ID DynamoDB (Allow):

< Services Actions in DynamoDB (4 of 65) Show remaining 61 actions

Read (1 of 26)

Action	Resource	Request condition
GetItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

Write (3 of 33)

Action	Resource	Request condition
DeleteItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}
PutItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}
UpdateItem	region| string like |us-west-1, TableName| string like |myDynamoTable	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

GetItem (Elenco) Riepilogo delle azioni:

< Actions GetItem action in DynamoDB

Resource	Region	Account	Request condition
region| string like |us-west-1, TableName| string like |myDynamoTable	us-west-1	123456789012	dynamodb:LeadingKeys = ${cognito-identity.amazonaws.com:sub}

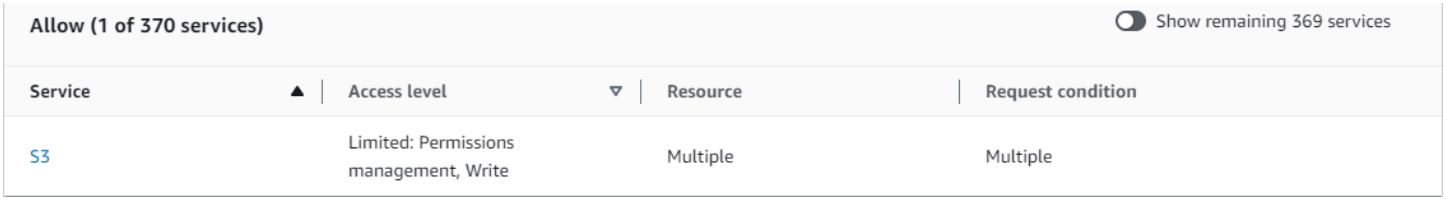
Politica 3: MultipleResourceCondition

Questa policy include più risorse e condizioni.

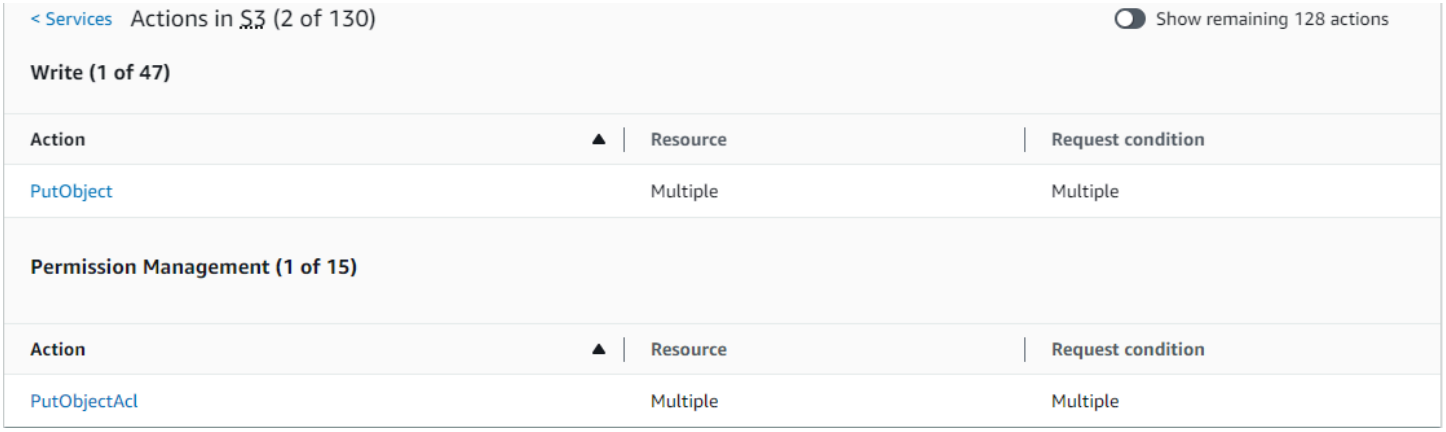
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Apple_bucket/*"],
            "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Orange_bucket/*"],
            "Condition": {"StringEquals": {
                "s3:x-amz-acl": ["custom"],
                "s3:x-amz-grant-full-control": ["1234"]
            }}
        }
    ]
}
```

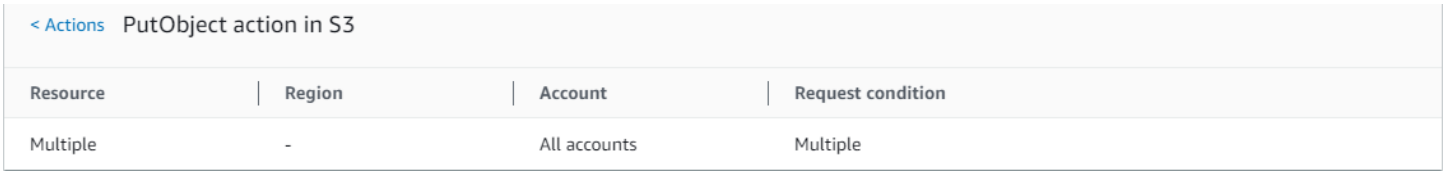
MultipleResourceConditionRiepilogo della politica:



MultipleResourceCondition Riepilogo del servizio S3 (Consenti):



PutObject (Scrivi) Riepilogo delle azioni:



Policy 4: EC2_troubleshoot

La policy seguente permette agli utenti di ottenere uno screenshot di un'istanza Amazon EC2 in esecuzione, che può essere utile per la risoluzione dei problemi di EC2. Questa policy permette inoltre di visualizzare le informazioni sugli elementi nel bucket degli sviluppatori di Amazon S3.

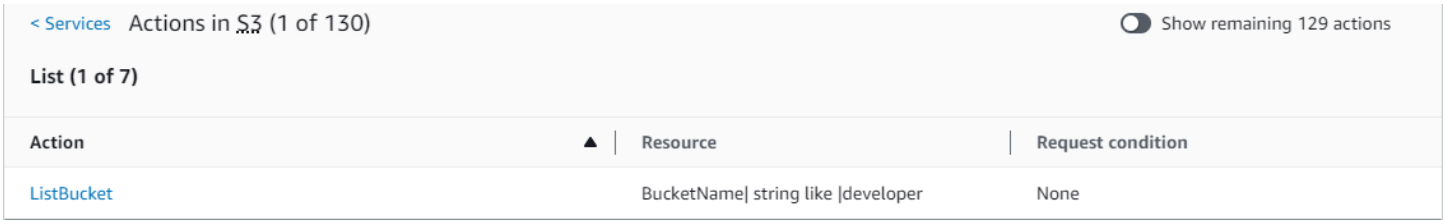
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshot"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::developer"
            ]
        }
    ]
}
```

Riepilogo della policy EC2_Troubleshoot:

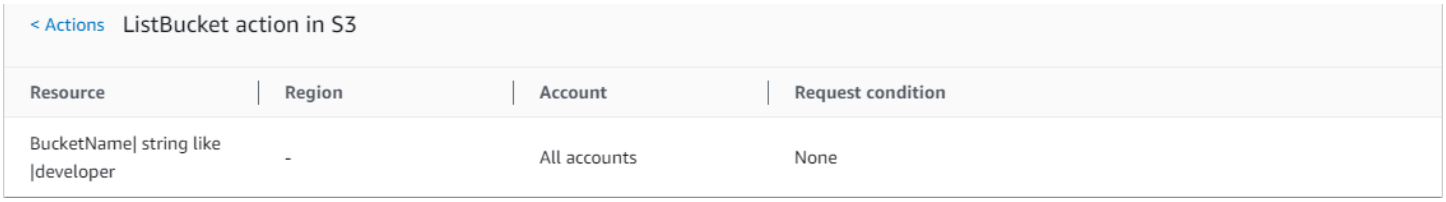
Allow (2 of 370 services) — Show remaining 368 services

Service	Access level	Resource	Request condition
EC2	Limited: Read	All resources	None
S3	Limited: List	BucketName| string like |developer	None

Riepilogo del servizio EC2_Troubleshoot S3 (permesso):



ListBucket (Elenco) Riepilogo delle azioni:

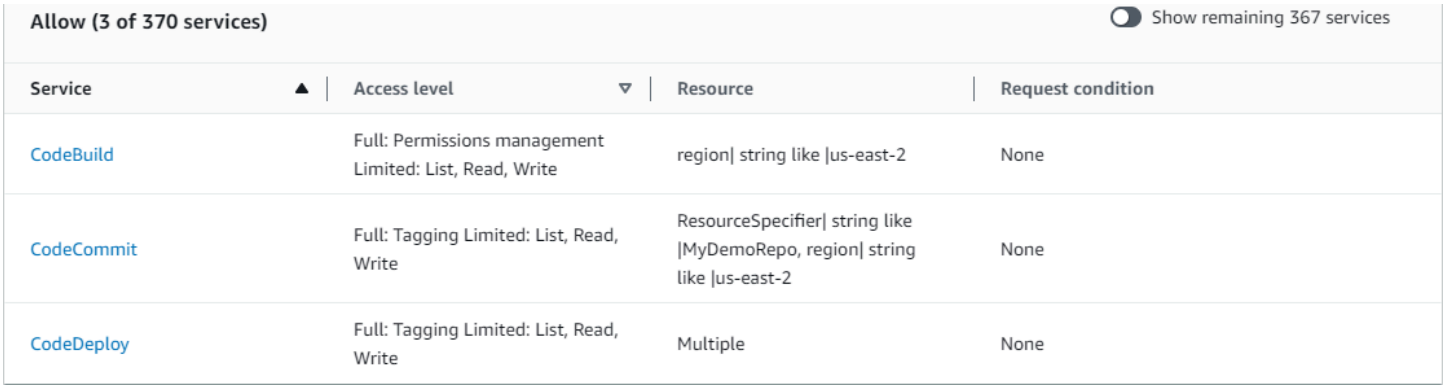


Politica 5: CodeBuild _ CodeCommit _ CodeDeploy

Questa politica fornisce l'accesso a CodeDeploy risorse e specifiche CodeBuild. CodeCommit Poiché queste risorse sono specifiche di ogni servizio, vengono visualizzate solo con il servizio corrispondente. Se includi una risorsa che non corrisponde ad alcun servizio nell'elemento `Action`, la risorsa viene visualizzata in tutti i riepiloghi delle operazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1487980617000",
            "Effect": "Allow",
            "Action": [
                "codebuild:*",
                "codecommit:*",
                "codedeploy:*"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project",
                "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
                "arn:aws:codedeploy:us-east-2:123456789012:application:WordPress_App",
                "arn:aws:codedeploy:us-east-2:123456789012:instance/AssetTag*"
            ]
        }
    ]
}
```

CodeBuild_ CodeCommit _ Riepilogo CodeDeploy della politica:



Allow (3 of 370 services) Show remaining 367 services

Service	Access level	Resource	Request condition
CodeBuild	Full: Permissions management Limited: List, Read, Write	region| string like |us-east-2	None
CodeCommit	Full: Tagging Limited: List, Read, Write	ResourceSpecifier| string like |MyDemoRepo, region| string like |us-east-2	None
CodeDeploy	Full: Tagging Limited: List, Read, Write	Multiple	None

CodeBuild_ CodeCommit _ CodeDeploy CodeBuild (Consenti) Riepilogo del servizio:

< Services Actions in CodeBuild (24 of 53) Show remaining 29 actions

Read (4 of 9)

Action	Resource	Request condition
BatchGetBuildBatches	region| string like |us-east-2	None
BatchGetBuilds	region| string like |us-east-2	None
BatchGetProjects	region| string like |us-east-2	None
GetResourcePolicy	region| string like |us-east-2	None

Write (16 of 28)

Action	Resource	Request condition
BatchDeleteBuilds	region| string like |us-east-2	None
CreateProject	region| string like |us-east-2	None
CreateWebhook	region| string like |us-east-2	None
DeleteBuildBatch	region| string like |us-east-2	None
DeleteProject	region| string like |us-east-2	None
DeleteWebhook	region| string like |us-east-2	None
InvalidateProjectCache	region| string like |us-east-2	None
RetryBuild	region| string like |us-east-2	None
RetryBuildBatch	region| string like |us-east-2	None
StartBuild	region| string like |us-east-2	None
StartBuildBatch	region| string like |us-east-2	None
StopBuild	region| string like |us-east-2	None
StopBuildBatch	region| string like |us-east-2	None
UpdateProject	region| string like |us-east-2	None
UpdateProjectVisibility	region| string like |us-east-2	None
UpdateWebhook	region| string like |us-east-2	None

List (2 of 14)

CodeBuild_ CodeCommit _ CodeDeploy StartBuild (Scrivi) Riepilogo delle azioni:

< Actions StartBuild action in CodeBuild

Resource	Region	Account	Request condition
region| string like |us-east-2	us-east-2	123456789012	None

Autorizzazioni necessarie per accedere alle risorse IAM

Le risorse sono oggetti all'interno di un servizio. Le risorse IAM includono gruppi, utenti, ruoli e policy. Se hai effettuato l'accesso con le credenziali dell' Utente root dell'account AWS non hai limitazioni per l'amministrazione delle credenziali IAM o delle risorse IAM. Tuttavia, agli utenti IAM devono essere esplicitamente concesse le autorizzazioni appropriate per amministrare credenziali o risorse IAM. A tale scopo, è possibile collegare all'utente una policy basata su identità.

> **Note**
>
> In tutta la AWS documentazione, quando facciamo riferimento a una policy IAM senza menzionare nessuna delle categorie specifiche, intendiamo una policy basata sull'identità e gestita dal cliente. Per ulteriori informazioni sulle categorie di policy, consultare the section called “Policy e autorizzazioni”.

Autorizzazioni per amministrare le identità IAM

Le autorizzazioni necessarie per amministrare gruppi, utenti, ruoli e credenziali IAM in genere corrispondono alle operazioni API per l'attività. Ad esempio, per creare utenti IAM, è necessario disporre dell'autorizzazione `iam:CreateUser` che corrisponde al comando API: `CreateUser`. Per consentire a un utente IAM di creare altri utenti IAM, è possibile collegare una policy AM come la seguente all'utente specificato:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

In una policy, il valore dell'elemento `Resource` dipende dall'operazione e da quali risorse possono essere interessate dall'operazione. Nell'esempio precedente, la policy consente a un utente di creare qualsiasi utente (`*` è un carattere jolly che corrisponde a tutte le stringhe). Per contro, una policy che consente agli utenti di modificare solo le proprie chiavi di accesso (operazioni API `CreateAccessKey` e `UpdateAccessKey`Resource) in genere include un elemento . In questo

caso l'ARN include una variabile (`${aws:username}`) che viene risolta nel nome dell'utente corrente, come nell'esempio seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListUsersForConsole",
            "Effect": "Allow",
            "Action": "iam:ListUsers",
            "Resource": "arn:aws:iam::*:*"
        },
        {
            "Sid": "ViewAndUpdateAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:UpdateAccessKey",
                "iam:CreateAccessKey",
                "iam:ListAccessKeys"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

Nell'esempio precedente, `${aws:username}` è una variabile che viene risulta nel nome utente dell'utente corrente. Per ulteriori informazioni sulle variabili di policy, consultare Elementi delle policy IAM: variabili e tag.

L'utilizzo di un carattere jolly (*) nel nome dell'operazione spesso facilita la concessione delle autorizzazioni per tutte le operazioni correlate a un'attività specifica. Ad esempio, per consentire agli utenti di eseguire qualsiasi operazione IAM, puoi utilizzare `iam:*` per l'operazione. Per consentire agli utenti di eseguire qualsiasi operazione correlata solo alle chiavi di accesso, puoi utilizzare `iam:*AccessKey*` nell'elemento `Action` di un'istruzione della policy. In questo modo l'utente ottiene l'autorizzazione per eseguire le operazioni `CreateAccessKey`, `DeleteAccessKey`, `GetAccessKeyLastUsed`, `ListAccessKeys` e `UpdateAccessKey`. (Se in futuro verrà aggiunta a IAM un'azione con "AccessKey" nel nome, l'utilizzo di `iam:*AccessKey*` for the `Action` element darà anche all'utente l'autorizzazione per quella nuova azione.) L'esempio seguente mostra una politica che consente agli utenti di eseguire tutte le azioni relative alle proprie chiavi di accesso (sostituirle *`account-id`* con il proprio Account AWS ID):

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::account-id:user/${aws:username}"
  }
}
```

Alcune attività, ad esempio l'eliminazione di un gruppo, coinvolgono più operazioni: devi prima rimuovere gli utenti dal gruppo, quindi scollegare o eliminare le policy del gruppo e quindi effettivamente eliminare il gruppo. Se desideri che un utente sia in grado di eliminare un gruppo, devi concedere all'utente le autorizzazioni necessarie per eseguire tutte le operazioni correlate.

Autorizzazioni per utilizzare la AWS Management Console

I precedenti esempi mostrano le policy che consentono a un utente di eseguire le operazioni con l'AWS CLIo gli SDK AWS.

Quando gli utenti utilizzano la console, questa emette richieste a IAM per elencare i gruppi, gli utenti, i ruoli e le policy e per ottenere le policy associate a un gruppo, un utente o un ruolo. La console invia anche richieste per ottenere Account AWS informazioni e informazioni sul principale. Il principale è l'utente che effettua le richieste nella console.

In generale, per eseguire un'operazione, è solo necessario che l'operazione corrispondente sia inclusa in una policy. Per creare un utente, è necessaria l'autorizzazione per chiamare l'operazione `CreateUser`. Spesso, quando utilizzi la console per eseguire un'operazione, devi disporre delle autorizzazioni per mostrare, elencare, ottenere o altrimenti visualizzare le risorse nella console. Ciò è necessario per poter navigare nella console allo scopo di eseguire l'operazione specificata. Ad esempio, se l'utente Jorge desidera utilizzare la console per modificare le proprie chiavi di accesso, passa alla console IAM e sceglie Utenti. Questa operazione determina l'esecuzione di una richiesta `ListUsers` da parte della console. Se Jorge non dispone dell'autorizzazione per l'operazione `iam:ListUsers`, alla console viene negato l'accesso quando cerca di elencare gli utenti. Di conseguenza, Jorge non può accedere al proprio nome e alle proprie chiavi di accesso, anche se dispone delle autorizzazioni per le operazioni `CreateAccessKey` e `UpdateAccessKey`.

Se desideri concedere agli utenti le autorizzazioni per amministrare gruppi, utenti, ruoli, politiche e credenziali con AWS Management Console, devi includere le autorizzazioni per le azioni eseguite

dalla console. Per alcuni esempi di policy che è possibile utilizzare per concedere a un utente tali autorizzazioni, consultare Esempi di policy per amministrare le risorse IAM.

Concessione di autorizzazioni tra account AWS

È possibile concedere direttamente agli utenti IAM dell'account l'accesso alle risorse. Se gli utenti di un altro account devono accedere alle risorse, è possibile creare un ruolo IAM ovvero un'entità che include le autorizzazioni, ma che non è associato a un utente specifico. Gli utenti di altri account possono utilizzare il ruolo e accedere alle risorse in base alle autorizzazioni assegnate al ruolo. Per ulteriori informazioni, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà.

> **Note**
>
> Alcuni servizi supportano le policy basate sulle risorse, come descritto in Policy basate sulle identità e policy basate su risorse (ad esempio Amazon S3, Amazon SNS e Amazon SQS). Per tali servizi, un'alternativa all'utilizzo dei ruoli consiste nel collegare una policy alla risorsa (bucket, argomento o coda) che si desidera condividere. La politica basata sulle risorse può specificare l' AWS account che dispone delle autorizzazioni per accedere alla risorsa.

Autorizzazioni per consentire a un servizio di accedere a un altro servizio

Molti AWS servizi accedono ad altri servizi. AWS Ad esempio, diversi AWS servizi, tra cui Amazon EMR, Elastic Load Balancing e Amazon EC2 Auto Scaling, gestiscono le istanze Amazon EC2. Altri AWS servizi utilizzano bucket Amazon S3, argomenti Amazon SNS, code Amazon SQS e così via.

Lo scenario per gestire le autorizzazioni in questi casi varia a seconda del servizio. Di seguito sono elencati alcuni esempi di come gestire le autorizzazioni per differenti servizi:

- In Amazon EC2 Auto Scaling, gli utenti devono disporre dell'autorizzazione per utilizzare Auto Scaling, ma non hanno bisogno dell'autorizzazione esplicita per gestire le istanze Amazon EC2.
- In effetti AWS Data Pipeline, un ruolo IAM determina cosa può fare una pipeline; gli utenti hanno bisogno dell'autorizzazione per assumere il ruolo. Per maggiori dettagli, consulta Concessione di autorizzazioni per le pipeline con IAM nella Guida per gli sviluppatori di AWS Data Pipeline .

Per dettagli su come configurare correttamente le autorizzazioni in modo che un AWS servizio sia in grado di svolgere le attività previste, consulta la documentazione del servizio che stai chiamando. Per

informazioni su come creare un ruolo per un servizio, consultare Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS.

Configurazione di un servizio con un ruolo IAM in modo che funzioni per tuo conto

Quando desideri configurare un AWS servizio in modo che funzioni per tuo conto, in genere fornisci l'ARN per un ruolo IAM che definisce ciò che il servizio è autorizzato a fare. AWS verifica di disporre delle autorizzazioni necessarie per trasferire un ruolo a un servizio. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS.

Operazioni necessarie

Le operazioni sono le azioni che puoi effettuare su una risorsa, come la visualizzazione, la creazione, la modifica e l'eliminazione di tale risorsa. Le azioni sono definite da ciascun AWS servizio.

Per consentire a qualcuno di eseguire un'operazione, è necessario includere le operazioni necessarie in una policy da applicare all'identità chiamante o alla risorsa interessata. In generale, per fornire le autorizzazioni necessarie per eseguire un'operazione, devi includere tale operazione nella policy. Ad esempio, per creare un utente, è necessario aggiungere l' CreateUser azione alla politica.

In alcuni casi, potrebbe essere necessario includere nella policy ulteriori azioni correlate per eseguire una determinata operazione. Ad esempio, per fornire a qualcuno l'autorizzazione per creare una directory in AWS Directory Service utilizzando l'operazione `ds:CreateDirectory`, devi includere le seguenti operazioni nella relativa policy:

- `ds:CreateDirectory`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:CreateSecurityGroup`
- `ec2:CreateNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:AuthorizeSecurityGroupEgress`

Quando crei o modifichi una policy utilizzando l'editor grafico, ricevi avvisi e richieste che ti aiutano a selezionare tutte le operazioni necessarie per la policy.

Per ulteriori informazioni sulle autorizzazioni necessarie per creare una directory in AWS Directory Service, vedi Esempio 2: Consentire a un utente di creare una directory.

Esempi di policy per amministrare le risorse IAM

Di seguito sono riportati gli esempi delle policy IAM che consentono agli utenti di eseguire attività associate alla gestione di utenti, gruppi e credenziali IAM. Queste includono le policy che consentono agli utenti di gestire le proprie password, le chiavi di accesso e i dispositivi per la multi-factor authentication (MFA).

Per esempi di policy che consentono agli utenti di eseguire attività con altri AWS servizi, come Amazon S3, Amazon EC2 e DynamoDB, consulta. Esempi di policy basate su identità IAM

Argomenti

- Consentire a un utente di elencare i gruppi, gli utenti e le policy dell'account e altro per scopi di report.
- Consentire a un utente di gestire l'appartenenza del gruppo
- Consentire a un utente di gestire gli utenti IAM
- Consentire agli utenti di impostare una policy per la password dell'account
- Consentire agli utenti di generare e recuperare i report delle credenziali IAM
- Consentire tutte le operazioni IAM (accesso amministratore)

Consentire a un utente di elencare i gruppi, gli utenti e le policy dell'account e altro per scopi di report.

La policy seguente consente all'utente di chiamare qualsiasi operazione IAM che inizi con la stringa `Get` o `List` e generare i report. Per visualizzare la policy di esempio, consulta IAM: consente l'accesso in sola lettura alla console IAM.

Consentire a un utente di gestire l'appartenenza del gruppo

La seguente politica consente all'utente di aggiornare l'appartenenza al gruppo chiamato. MarketingGroup Per visualizzare la policy di esempio, consulta IAM: consente di gestire l'appartenenza di un gruppo a livello di programmazione e nella console.

Consentire a un utente di gestire gli utenti IAM

La policy seguente consente a un utente di eseguire tutte le attività associate alla gestione degli utenti IAM ma non di eseguire le operazioni su altre entità, come ad esempio la creazione di gruppi o policy. Le operazioni consentite includono le seguenti:

- Creazione dell'utente (l'operazione `CreateUser`).
- Eliminazione dell'utente. Questa operazione richiede le autorizzazioni per eseguire tutte le seguenti operazioni: `DeleteSigningCertificate`, `DeleteLoginProfile`, `RemoveUserFromGroup` e `DeleteUser`.
- Elencare gli utenti nell'account e nei gruppi (le operazioni `GetUser`, `ListUsers` e `ListGroupsForUser`).
- Elencare e rimuovere le policy per l'utente (le operazioni `ListUserPolicies`, `ListAttachedUserPolicies`, `DetachUserPolicy`, `DeleteUserPolicy`)
- Rinominare o modificare il percorso per l'utente (l'operazione `UpdateUser`). L'elemento `Resource` deve includere un ARN che copre sia il percorso di origine sia il percorso di destinazione. Per ulteriori informazioni sui percorsi, consultare la pagina Nomi descrittivi e percorsi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUsersToPerformUserActions",
            "Effect": "Allow",
            "Action": [
                "iam:ListPolicies",
                "iam:GetPolicy",
                "iam:UpdateUser",
                "iam:AttachUserPolicy",
                "iam:ListEntitiesForPolicy",
                "iam:DeleteUserPolicy",
                "iam:DeleteUser",
                "iam:ListUserPolicies",
                "iam:CreateUser",
                "iam:RemoveUserFromGroup",
                "iam:AddUserToGroup",
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:PutUserPolicy",
```

```
                "iam:ListAttachedUserPolicies",
                "iam:ListUsers",
                "iam:GetUser",
                "iam:DetachUserPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUsersToSeeStatsOnIAMConsoleDashboard",
            "Effect": "Allow",
            "Action": [
                "iam:GetAccount*",
                "iam:ListAccount*"
            ],
            "Resource": "*"
        }
    ]
}
```

Un numero di autorizzazioni incluse nella policy precedente consente all'utente di eseguire attività nella AWS Management Console. Gli utenti che eseguono attività correlate all'utente dalla AWS CLI, gli SDK AWS o dalle API di query HTTP IAM potrebbero non avere bisogno di determinate autorizzazioni. Ad esempio, se gli utenti conoscono già l'ARN delle policy per distaccarsi da un utente, non hanno bisogno dell'autorizzazione `iam:ListAttachedUserPolicies`. L'elenco esatto delle autorizzazioni che un utente richiede, dipende dalle attività che l'utente deve eseguire durante la gestione di altri utenti.

I seguenti permessi nella policy consentono l'accesso alle attività dell'utente tramite la AWS Management Console:

- `iam:GetAccount*`
- `iam:ListAccount*`

Consentire agli utenti di impostare una policy per la password dell'account

Potresti fornire ad alcuni utenti le autorizzazioni per ottenere e aggiornare la password policy (policy della password) del tuo Account AWS. Per visualizzare la policy di esempio, consulta IAM: consente l'impostazione dei requisiti della password dell'account a livello di programmazione e nella console.

Consentire agli utenti di generare e recuperare i report delle credenziali IAM

Puoi concedere agli utenti il permesso di generare e scaricare un rapporto che elenca tutti gli utenti del tuo Account AWS. Il report elenca inoltre lo stato di varie credenziali utente, tra cui le password, le chiavi di accesso, i dispositivi MFA e i certificati di firma. Per ulteriori informazioni sui report delle credenziali, consultare la pagina Recupero dei report delle credenziali per l' Account AWS. Per visualizzare la policy di esempio, consulta IAM: generazione e recupero di report di credenziali IAM.

Consentire tutte le operazioni IAM (accesso amministratore)

È possibile fornire ad alcuni utenti le autorizzazioni amministrative per eseguire tutte le operazioni in IAM, tra cui la gestione delle password, le chiavi di accesso, i dispositivi MFA e i certificati utente. Il seguente esempio di policy concede queste autorizzazioni:

> **Warning**
>
> Quando concedi a un utente l'accesso completo a IAM, non esiste alcun limite alle autorizzazioni che l'utente può concedere a se stesso o agli altri. L'utente può creare nuove entità IAM (utenti o ruoli) e concedere a quelle entità l'accesso completo a tutte le risorse nel tuo Account AWS. Quando concedi a un utente l'accesso completo a IAM, stai concedendo effettivamente l'accesso completo a tutte le risorse nel tuo Account AWS. Questo include l'accesso a eliminare tutte le risorse. Dovresti concedere queste autorizzazioni solo agli amministratori attendibili e dovresti applicare la multi-factor authentication (MFA) per questi amministratori.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*",
    "Resource": "*"
  }
}
```

Esempi di codice per IAM che utilizza gli AWS SDK

I seguenti esempi di codice mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Utilizzo DeleteAccessKey con un AWS SDK o una CLI
- Utilizzo DeleteAccountAlias con un AWS SDK o una CLI
- Utilizzo DeleteAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo DeleteGroup con un AWS SDK o una CLI
- Utilizzo DeleteGroupPolicy con un AWS SDK o una CLI
- Utilizzo DeleteInstanceProfile con un AWS SDK o una CLI
- Utilizzo DeleteLoginProfile con un AWS SDK o una CLI
- Utilizzo DeleteOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo DeletePolicy con un AWS SDK o una CLI
- Utilizzo DeletePolicyVersion con un AWS SDK o una CLI
- Utilizzo DeleteRole con un AWS SDK o una CLI
- Utilizzo DeleteRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteRolePolicy con un AWS SDK o una CLI
- Utilizzo DeleteSAMLProvider con un AWS SDK o una CLI
- Utilizzo DeleteServerCertificate con un AWS SDK o una CLI
- Utilizzo DeleteServiceLinkedRole con un AWS SDK o una CLI
- Utilizzo DeleteSigningCertificate con un AWS SDK o una CLI
- Utilizzo DeleteUser con un AWS SDK o una CLI
- Utilizzo DeleteUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteUserPolicy con un AWS SDK o una CLI
- Utilizzo DeleteVirtualMfaDevice con un AWS SDK o una CLI
- Utilizzo DetachGroupPolicy con un AWS SDK o una CLI
- Utilizzo DetachRolePolicy con un AWS SDK o una CLI
- Utilizzo DetachUserPolicy con un AWS SDK o una CLI
- Utilizzo EnableMfaDevice con un AWS SDK o una CLI
- Utilizzo GenerateCredentialReport con un AWS SDK o una CLI
- Utilizzo GenerateServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetAccessKeyLastUsed con un AWS SDK o una CLI
- Utilizzo GetAccountAuthorizationDetails con un AWS SDK o una CLI
- Utilizzo GetAccountPasswordPolicy con un AWS SDK o una CLI

- Utilizzo GetAccountSummary con un AWS SDK o una CLI
- Utilizzo GetContextKeysForCustomPolicy con un AWS SDK o una CLI
- Utilizzo GetContextKeysForPrincipalPolicy con un AWS SDK o una CLI
- Utilizzo GetCredentialReport con un AWS SDK o una CLI
- Utilizzo GetGroup con un AWS SDK o una CLI
- Utilizzo GetGroupPolicy con un AWS SDK o una CLI
- Utilizzo GetInstanceProfile con un AWS SDK o una CLI
- Utilizzo GetLoginProfile con un AWS SDK o una CLI
- Utilizzo GetOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo GetPolicy con un AWS SDK o una CLI
- Utilizzo GetPolicyVersion con un AWS SDK o una CLI
- Utilizzo GetRole con un AWS SDK o una CLI
- Utilizzo GetRolePolicy con un AWS SDK o una CLI
- Utilizzo GetSamlProvider con un AWS SDK o una CLI
- Utilizzo GetServerCertificate con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetailsWithEntities con un AWS SDK o una CLI
- Utilizzo GetServiceLinkedRoleDeletionStatus con un AWS SDK o una CLI
- Utilizzo GetUser con un AWS SDK o una CLI
- Utilizzo GetUserPolicy con un AWS SDK o una CLI
- Utilizzo ListAccessKeys con un AWS SDK o una CLI
- Utilizzo ListAccountAliases con un AWS SDK o una CLI
- Utilizzo ListAttachedGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedRolePolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedUserPolicies con un AWS SDK o una CLI
- Utilizzo ListEntitiesForPolicy con un AWS SDK o una CLI
- Utilizzo ListGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListGroups con un AWS SDK o una CLI
- Utilizzo ListGroupsForUser con un AWS SDK o una CLI
- Utilizzo ListInstanceProfiles con un AWS SDK o una CLI

- Utilizzo ListInstanceProfilesForRole con un AWS SDK o una CLI
- Utilizzo ListMfaDevices con un AWS SDK o una CLI
- Utilizzo ListOpenIdConnectProviders con un AWS SDK o una CLI
- Utilizzo ListPolicies con un AWS SDK o una CLI
- Utilizzo ListPolicyVersions con un AWS SDK o una CLI
- Utilizzo ListRolePolicies con un AWS SDK o una CLI
- Utilizzo ListRoleTags con un AWS SDK o una CLI
- Utilizzo ListRoles con un AWS SDK o una CLI
- Utilizzo ListSAMLProviders con un AWS SDK o una CLI
- Utilizzo ListServerCertificates con un AWS SDK o una CLI
- Utilizzo ListSigningCertificates con un AWS SDK o una CLI
- Utilizzo ListUserPolicies con un AWS SDK o una CLI
- Utilizzo ListUserTags con un AWS SDK o una CLI
- Utilizzo ListUsers con un AWS SDK o una CLI
- Utilizzo ListVirtualMfaDevices con un AWS SDK o una CLI
- Utilizzo PutGroupPolicy con un AWS SDK o una CLI
- Utilizzo PutRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutRolePolicy con un AWS SDK o una CLI
- Utilizzo PutUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutUserPolicy con un AWS SDK o una CLI
- Utilizzo RemoveClientIdFromOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo RemoveRoleFromInstanceProfile con un AWS SDK o una CLI
- Utilizzo RemoveUserFromGroup con un AWS SDK o una CLI
- Utilizzo ResyncMfaDevice con un AWS SDK o una CLI
- Utilizzo SetDefaultPolicyVersion con un AWS SDK o una CLI
- Utilizzo TagRole con un AWS SDK o una CLI
- Utilizzo TagUser con un AWS SDK o una CLI
- Utilizzo UntagRole con un AWS SDK o una CLI
- Utilizzo UntagUser con un AWS SDK o una CLI
- Utilizzo UpdateAccessKey con un AWS SDK o una CLI

    - Utilizzo GetSessionToken con un AWS SDK o una CLI
- Scenari per l' AWS STS utilizzo degli AWS SDK
    - Assumi un ruolo IAM che richiede un token MFA con l' AWS STS utilizzo di un SDK AWS
    - Costruisci un URL con AWS STS per utenti federati utilizzando un SDK AWS
    - Ottieni un token di sessione che richiede un token MFA AWS STS utilizzando un SDK AWS

Esempi di codice per IAM che utilizza gli AWS SDK

I seguenti esempi di codice mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Hello IAM

Gli esempi di codice seguenti mostrano come iniziare a utilizzare IAM.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace IAMActions;
```

```
public class HelloIAM
{
    static async Task Main(string[] args)
    {
        // Getting started with AWS Identity and Access Management (IAM). List
        // the policies for the account.
        var iamClient = new AmazonIdentityManagementServiceClient();

        var listPoliciesPaginator = iamClient.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        Console.WriteLine("Here are the policies defined for your account:\n");
        policies.ForEach(policy =>
        {
            Console.WriteLine($"Created:
 {policy.CreateDate}\t{policy.PolicyName}\t{policy.Description}");
        });
    }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per il file CMake C MakeLists .txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS iam)

# Set this project's name.
project("hello_iam")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
    # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

    # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
 may need to uncomment this
    # and set the proper subdirectory to the executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_iam.cpp)

target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Codice per il file origine iam.cpp.

```
#include <aws/core/Aws.h>
#include <aws/iam/IAMClient.h>
#include <aws/iam/model/ListPoliciesRequest.h>
#include <iostream>
#include <iomanip>

/*
 *  A "Hello IAM" starter application which initializes an AWS Identity and
 Access Management (IAM) client
 *  and lists the IAM policies.
 *
 *  main function
 *
 *  Usage: 'hello_iam'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        const Aws::String DATE_FORMAT("%Y-%m-%d");
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::IAM::IAMClient iamClient(clientConfig);
        Aws::IAM::Model::ListPoliciesRequest request;

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = iamClient.ListPolicies(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Failed to list iam policies: " <<
                          outcome.GetError().GetMessage() << std::endl;
```

```
                result = 1;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(55) << "Name" <<
                          std::setw(30) << "ID" << std::setw(80) << "Arn" <<
                          std::setw(64) << "Description" << std::setw(12) <<
                          "CreateDate" << std::endl;
                header = true;
            }

            const auto &policies = outcome.GetResult().GetPolicies();
            for (const auto &policy: policies) {
                std::cout << std::left << std::setw(55) <<
                          policy.GetPolicyName() << std::setw(30) <<
                          policy.GetPolicyId() << std::setw(80) <<
 policy.GetArn() <<
                          std::setw(64) << policy.GetDescription() <<
 std::setw(12) <<
                          policy.GetCreateDate().ToGmtString(DATE_FORMAT.c_str())
 <<
                          std::endl;
            }

            if (outcome.GetResult().GetIsTruncated()) {
                request.SetMarker(outcome.GetResult().GetMarker());
            } else {
                done = true;
            }
        }
    }


    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Per i dettagli sull'API, consulta la sezione API ListPoliciesReference AWS SDK for C++ .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/iam"
)

// main uses the AWS SDK for Go (v2) to create an AWS Identity and Access
 Management (IAM)
// client and list up to 10 policies in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 iamClient := iam.NewFromConfig(sdkConfig)
 const maxPols = 10
 fmt.Printf("Let's list up to %v policies for your account.\n", maxPols)
 result, err := iamClient.ListPolicies(context.TODO(), &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPols),
 })
 if err != nil {
```

```
  fmt.Printf("Couldn't list policies for your account. Here's why: %v\n", err)
  return
 }
 if len(result.Policies) == 0 {
  fmt.Println("You don't have any policies!")
 } else {
  for _, policy := range result.Policies {
   fmt.Printf("\t%v\n", *policy.PolicyName)
  }
 }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.ListPoliciesResponse;
import software.amazon.awssdk.services.iam.model.Policy;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloIAM {
```

```
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listPolicies(iam);
    }

    public static void listPolicies(IamClient iam) {
        ListPoliciesResponse response = iam.listPolicies();
        List<Policy> polList = response.policies();
        polList.forEach(policy -> {
            System.out.println("Policy Name: " + policy.policyName());
        });
    }
}
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { IAMClient, paginateListPolicies } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listLocalPolicies = async () => {
  /**
   * In v3, the clients expose paginateOperationName APIs that are written using
 async generators so that you can use async iterators in a for await..of loop.
   * https://docs.aws.amazon.com/AWSJavaScriptSDK/v3/latest/index.html#paginators
   */
  const paginator = paginateListPolicies(
```

```
    { client, pageSize: 10 },
    // List only customer managed policies.
    { Scope: "Local" },
  );

  console.log("IAM policies defined in your account:");
  let policyCount = 0;
  for await (const page of paginator) {
    if (page.Policies) {
      page.Policies.forEach((p) => {
        console.log(`${p.PolicyName}`);
        policyCount++;
      });
    }
  }
  console.log(`Found ${policyCount} policies.`);
};
```

- Per i dettagli sull'API, consulta la ListPoliciessezione AWS SDK for JavaScript API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Da src/bin/hello.rs.

```
use aws_sdk_iam::error::SdkError;
use aws_sdk_iam::operation::list_policies::ListPoliciesError;
use clap::Parser;

const PATH_PREFIX_HELP: &str = "The path prefix for filtering the results.";

#[derive(Debug, clap::Parser)]
```

```
#[command(about)]
struct HelloScenarioArgs {
    #[arg(long, default_value="/", help=PATH_PREFIX_HELP)]
    pub path_prefix: String,
}

#[tokio::main]
async fn main() -> Result<(), SdkError<ListPoliciesError>> {
    let sdk_config = aws_config::load_from_env().await;
    let client = aws_sdk_iam::Client::new(&sdk_config);

    let args = HelloScenarioArgs::parse();

    iam_service::list_policies(client, args.path_prefix).await?;

    Ok(())
}
```

Da src/iam-service-lib.rs.

```
pub async fn list_policies(
    client: iamClient,
    path_prefix: String,
) -> Result<Vec<String>, SdkError<ListPoliciesError>> {
    let list_policies = client
        .list_policies()
        .path_prefix(path_prefix)
        .scope(PolicyScopeType::Local)
        .into_paginator()
        .items()
        .send()
        .try_collect()
        .await?;

    let policy_names = list_policies
        .into_iter()
        .map(|p| {
            let name = p
                .policy_name
                .unwrap_or_else(|| "Missing Policy Name".to_string());
            println!("{}", name);
            name
```

```
        })
        .collect();

    Ok(policy_names)
}
```

- Per i dettagli sulle API, consulta la ListPoliciesguida di riferimento all'API AWS SDK for Rust.

Esempi di codice

- Azioni per IAM che utilizzano gli AWS SDK
  - Utilizzo AddClientIdToOpenIdConnectProvider con un AWS SDK o una CLI
  - Utilizzo AddRoleToInstanceProfile con un AWS SDK o una CLI
  - Utilizzo AddUserToGroup con un AWS SDK o una CLI
  - Utilizzo AttachGroupPolicy con un AWS SDK o una CLI
  - Utilizzo AttachRolePolicy con un AWS SDK o una CLI
  - Utilizzo AttachUserPolicy con un AWS SDK o una CLI
  - Utilizzo ChangePassword con un AWS SDK o una CLI
  - Utilizzo CreateAccessKey con un AWS SDK o una CLI
  - Utilizzo CreateAccountAlias con un AWS SDK o una CLI
  - Utilizzo CreateGroup con un AWS SDK o una CLI
  - Utilizzo CreateInstanceProfile con un AWS SDK o una CLI
  - Utilizzo CreateLoginProfile con un AWS SDK o una CLI
  - Utilizzo CreateOpenIdConnectProvider con un AWS SDK o una CLI
  - Utilizzo CreatePolicy con un AWS SDK o una CLI
  - Utilizzo CreatePolicyVersion con un AWS SDK o una CLI
  - Utilizzo CreateRole con un AWS SDK o una CLI
  - Utilizzo CreateSAMLProvider con un AWS SDK o una CLI
  - Utilizzo CreateServiceLinkedRole con un AWS SDK o una CLI
  - Utilizzo CreateUser con un AWS SDK o una CLI
  - Utilizzo CreateVirtualMfaDevice con un AWS SDK o una CLI
  - Utilizzo DeactivateMfaDevice con un AWS SDK o una CLI
  - Utilizzo DeleteAccessKey con un AWS SDK o una CLI

- Utilizzo DeleteAccountAlias con un AWS SDK o una CLI
- Utilizzo DeleteAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo DeleteGroup con un AWS SDK o una CLI
- Utilizzo DeleteGroupPolicy con un AWS SDK o una CLI
- Utilizzo DeleteInstanceProfile con un AWS SDK o una CLI
- Utilizzo DeleteLoginProfile con un AWS SDK o una CLI
- Utilizzo DeleteOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo DeletePolicy con un AWS SDK o una CLI
- Utilizzo DeletePolicyVersion con un AWS SDK o una CLI
- Utilizzo DeleteRole con un AWS SDK o una CLI
- Utilizzo DeleteRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteRolePolicy con un AWS SDK o una CLI
- Utilizzo DeleteSAMLProvider con un AWS SDK o una CLI
- Utilizzo DeleteServerCertificate con un AWS SDK o una CLI
- Utilizzo DeleteServiceLinkedRole con un AWS SDK o una CLI
- Utilizzo DeleteSigningCertificate con un AWS SDK o una CLI
- Utilizzo DeleteUser con un AWS SDK o una CLI
- Utilizzo DeleteUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteUserPolicy con un AWS SDK o una CLI
- Utilizzo DeleteVirtualMfaDevice con un AWS SDK o una CLI
- Utilizzo DetachGroupPolicy con un AWS SDK o una CLI
- Utilizzo DetachRolePolicy con un AWS SDK o una CLI
- Utilizzo DetachUserPolicy con un AWS SDK o una CLI
- Utilizzo EnableMfaDevice con un AWS SDK o una CLI
- Utilizzo GenerateCredentialReport con un AWS SDK o una CLI
- Utilizzo GenerateServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetAccessKeyLastUsed con un AWS SDK o una CLI
- Utilizzo GetAccountAuthorizationDetails con un AWS SDK o una CLI
- Utilizzo GetAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo GetAccountSummary con un AWS SDK o una CLI

- Utilizzo GetContextKeysForCustomPolicy con un AWS SDK o una CLI
- Utilizzo GetContextKeysForPrincipalPolicy con un AWS SDK o una CLI
- Utilizzo GetCredentialReport con un AWS SDK o una CLI
- Utilizzo GetGroup con un AWS SDK o una CLI
- Utilizzo GetGroupPolicy con un AWS SDK o una CLI
- Utilizzo GetInstanceProfile con un AWS SDK o una CLI
- Utilizzo GetLoginProfile con un AWS SDK o una CLI
- Utilizzo GetOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo GetPolicy con un AWS SDK o una CLI
- Utilizzo GetPolicyVersion con un AWS SDK o una CLI
- Utilizzo GetRole con un AWS SDK o una CLI
- Utilizzo GetRolePolicy con un AWS SDK o una CLI
- Utilizzo GetSamlProvider con un AWS SDK o una CLI
- Utilizzo GetServerCertificate con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetailsWithEntities con un AWS SDK o una CLI
- Utilizzo GetServiceLinkedRoleDeletionStatus con un AWS SDK o una CLI
- Utilizzo GetUser con un AWS SDK o una CLI
- Utilizzo GetUserPolicy con un AWS SDK o una CLI
- Utilizzo ListAccessKeys con un AWS SDK o una CLI
- Utilizzo ListAccountAliases con un AWS SDK o una CLI
- Utilizzo ListAttachedGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedRolePolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedUserPolicies con un AWS SDK o una CLI
- Utilizzo ListEntitiesForPolicy con un AWS SDK o una CLI
- Utilizzo ListGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListGroups con un AWS SDK o una CLI
- Utilizzo ListGroupsForUser con un AWS SDK o una CLI
- Utilizzo ListInstanceProfiles con un AWS SDK o una CLI
- Utilizzo ListInstanceProfilesForRole con un AWS SDK o una CLI

- Utilizzo ListMfaDevices con un AWS SDK o una CLI
- Utilizzo ListOpenIdConnectProviders con un AWS SDK o una CLI
- Utilizzo ListPolicies con un AWS SDK o una CLI
- Utilizzo ListPolicyVersions con un AWS SDK o una CLI
- Utilizzo ListRolePolicies con un AWS SDK o una CLI
- Utilizzo ListRoleTags con un AWS SDK o una CLI
- Utilizzo ListRoles con un AWS SDK o una CLI
- Utilizzo ListSAMLProviders con un AWS SDK o una CLI
- Utilizzo ListServerCertificates con un AWS SDK o una CLI
- Utilizzo ListSigningCertificates con un AWS SDK o una CLI
- Utilizzo ListUserPolicies con un AWS SDK o una CLI
- Utilizzo ListUserTags con un AWS SDK o una CLI
- Utilizzo ListUsers con un AWS SDK o una CLI
- Utilizzo ListVirtualMfaDevices con un AWS SDK o una CLI
- Utilizzo PutGroupPolicy con un AWS SDK o una CLI
- Utilizzo PutRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutRolePolicy con un AWS SDK o una CLI
- Utilizzo PutUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutUserPolicy con un AWS SDK o una CLI
- Utilizzo RemoveClientIdFromOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo RemoveRoleFromInstanceProfile con un AWS SDK o una CLI
- Utilizzo RemoveUserFromGroup con un AWS SDK o una CLI
- Utilizzo ResyncMfaDevice con un AWS SDK o una CLI
- Utilizzo SetDefaultPolicyVersion con un AWS SDK o una CLI
- Utilizzo TagRole con un AWS SDK o una CLI
- Utilizzo TagUser con un AWS SDK o una CLI
- Utilizzo UntagRole con un AWS SDK o una CLI
- Utilizzo UntagUser con un AWS SDK o una CLI
- Utilizzo UpdateAccessKey con un AWS SDK o una CLI

- Utilizzo UpdateAccountPasswordPolicy con un AWS SDK o una CLI

- Utilizzo UpdateAssumeRolePolicy con un AWS SDK o una CLI
- Utilizzo UpdateGroup con un AWS SDK o una CLI
- Utilizzo UpdateLoginProfile con un AWS SDK o una CLI
- Utilizzo UpdateOpenIdConnectProviderThumbprint con un AWS SDK o una CLI
- Utilizzo UpdateRole con un AWS SDK o una CLI
- Utilizzo UpdateRoleDescription con un AWS SDK o una CLI
- Utilizzo UpdateSamlProvider con un AWS SDK o una CLI
- Utilizzo UpdateServerCertificate con un AWS SDK o una CLI
- Utilizzo UpdateSigningCertificate con un AWS SDK o una CLI
- Utilizzo UpdateUser con un AWS SDK o una CLI
- Utilizzo UploadServerCertificate con un AWS SDK o una CLI
- Utilizzo UploadSigningCertificate con un AWS SDK o una CLI

- Scenari per IAM che utilizzano AWS SDK
  - Crea e gestisci un servizio resiliente utilizzando un SDK AWS
  - Crea un gruppo IAM e aggiungi un utente al gruppo utilizzando un AWS SDK
  - Crea un utente IAM e assumi un ruolo AWS STS utilizzando un AWS SDK
  - Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un SDK AWS
  - Gestisci le chiavi di accesso IAM utilizzando un AWS SDK
  - Gestisci le policy IAM utilizzando un AWS SDK
  - Gestisci i ruoli IAM utilizzando un AWS SDK
  - Gestisci il tuo account IAM utilizzando un AWS SDK
  - Ripristina una versione della policy IAM utilizzando un AWS SDK
  - Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

Azioni per IAM che utilizzano gli AWS SDK

I seguenti esempi di codice mostrano come eseguire singole azioni IAM con gli AWS SDK. Questi estratti chiamano l'API IAM e sono estratti di codice da programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la Documentazione di riferimento delle API AWS Identity and Access Management (IAM).

Esempi

- Utilizzo AddClientIdToOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo AddRoleToInstanceProfile con un AWS SDK o una CLI
- Utilizzo AddUserToGroup con un AWS SDK o una CLI
- Utilizzo AttachGroupPolicy con un AWS SDK o una CLI
- Utilizzo AttachRolePolicy con un AWS SDK o una CLI
- Utilizzo AttachUserPolicy con un AWS SDK o una CLI
- Utilizzo ChangePassword con un AWS SDK o una CLI
- Utilizzo CreateAccessKey con un AWS SDK o una CLI
- Utilizzo CreateAccountAlias con un AWS SDK o una CLI
- Utilizzo CreateGroup con un AWS SDK o una CLI
- Utilizzo CreateInstanceProfile con un AWS SDK o una CLI
- Utilizzo CreateLoginProfile con un AWS SDK o una CLI
- Utilizzo CreateOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo CreatePolicy con un AWS SDK o una CLI
- Utilizzo CreatePolicyVersion con un AWS SDK o una CLI
- Utilizzo CreateRole con un AWS SDK o una CLI
- Utilizzo CreateSAMLProvider con un AWS SDK o una CLI
- Utilizzo CreateServiceLinkedRole con un AWS SDK o una CLI
- Utilizzo CreateUser con un AWS SDK o una CLI
- Utilizzo CreateVirtualMfaDevice con un AWS SDK o una CLI
- Utilizzo DeactivateMfaDevice con un AWS SDK o una CLI
- Utilizzo DeleteAccessKey con un AWS SDK o una CLI
- Utilizzo DeleteAccountAlias con un AWS SDK o una CLI
- Utilizzo DeleteAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo DeleteGroup con un AWS SDK o una CLI

- Utilizzo DeleteGroupPolicy con un AWS SDK o una CLI
- Utilizzo DeleteInstanceProfile con un AWS SDK o una CLI
- Utilizzo DeleteLoginProfile con un AWS SDK o una CLI
- Utilizzo DeleteOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo DeletePolicy con un AWS SDK o una CLI
- Utilizzo DeletePolicyVersion con un AWS SDK o una CLI
- Utilizzo DeleteRole con un AWS SDK o una CLI
- Utilizzo DeleteRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteRolePolicy con un AWS SDK o una CLI
- Utilizzo DeleteSAMLProvider con un AWS SDK o una CLI
- Utilizzo DeleteServerCertificate con un AWS SDK o una CLI
- Utilizzo DeleteServiceLinkedRole con un AWS SDK o una CLI
- Utilizzo DeleteSigningCertificate con un AWS SDK o una CLI
- Utilizzo DeleteUser con un AWS SDK o una CLI
- Utilizzo DeleteUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo DeleteUserPolicy con un AWS SDK o una CLI
- Utilizzo DeleteVirtualMfaDevice con un AWS SDK o una CLI
- Utilizzo DetachGroupPolicy con un AWS SDK o una CLI
- Utilizzo DetachRolePolicy con un AWS SDK o una CLI
- Utilizzo DetachUserPolicy con un AWS SDK o una CLI
- Utilizzo EnableMfaDevice con un AWS SDK o una CLI
- Utilizzo GenerateCredentialReport con un AWS SDK o una CLI
- Utilizzo GenerateServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetAccessKeyLastUsed con un AWS SDK o una CLI
- Utilizzo GetAccountAuthorizationDetails con un AWS SDK o una CLI
- Utilizzo GetAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo GetAccountSummary con un AWS SDK o una CLI
- Utilizzo GetContextKeysForCustomPolicy con un AWS SDK o una CLI

- Utilizzo GetContextKeysForPrincipalPolicy con un AWS SDK o una CLI
- Utilizzo GetCredentialReport con un AWS SDK o una CLI
- Utilizzo GetGroup con un AWS SDK o una CLI
- Utilizzo GetGroupPolicy con un AWS SDK o una CLI
- Utilizzo GetInstanceProfile con un AWS SDK o una CLI
- Utilizzo GetLoginProfile con un AWS SDK o una CLI
- Utilizzo GetOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo GetPolicy con un AWS SDK o una CLI
- Utilizzo GetPolicyVersion con un AWS SDK o una CLI
- Utilizzo GetRole con un AWS SDK o una CLI
- Utilizzo GetRolePolicy con un AWS SDK o una CLI
- Utilizzo GetSamlProvider con un AWS SDK o una CLI
- Utilizzo GetServerCertificate con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetails con un AWS SDK o una CLI
- Utilizzo GetServiceLastAccessedDetailsWithEntities con un AWS SDK o una CLI
- Utilizzo GetServiceLinkedRoleDeletionStatus con un AWS SDK o una CLI
- Utilizzo GetUser con un AWS SDK o una CLI
- Utilizzo GetUserPolicy con un AWS SDK o una CLI
- Utilizzo ListAccessKeys con un AWS SDK o una CLI
- Utilizzo ListAccountAliases con un AWS SDK o una CLI
- Utilizzo ListAttachedGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedRolePolicies con un AWS SDK o una CLI
- Utilizzo ListAttachedUserPolicies con un AWS SDK o una CLI
- Utilizzo ListEntitiesForPolicy con un AWS SDK o una CLI
- Utilizzo ListGroupPolicies con un AWS SDK o una CLI
- Utilizzo ListGroups con un AWS SDK o una CLI
- Utilizzo ListGroupsForUser con un AWS SDK o una CLI
- Utilizzo ListInstanceProfiles con un AWS SDK o una CLI

- Utilizzo ListInstanceProfilesForRole con un AWS SDK o una CLI
- Utilizzo ListMfaDevices con un AWS SDK o una CLI
- Utilizzo ListOpenIdConnectProviders con un AWS SDK o una CLI
- Utilizzo ListPolicies con un AWS SDK o una CLI
- Utilizzo ListPolicyVersions con un AWS SDK o una CLI
- Utilizzo ListRolePolicies con un AWS SDK o una CLI
- Utilizzo ListRoleTags con un AWS SDK o una CLI
- Utilizzo ListRoles con un AWS SDK o una CLI
- Utilizzo ListSAMLProviders con un AWS SDK o una CLI
- Utilizzo ListServerCertificates con un AWS SDK o una CLI
- Utilizzo ListSigningCertificates con un AWS SDK o una CLI
- Utilizzo ListUserPolicies con un AWS SDK o una CLI
- Utilizzo ListUserTags con un AWS SDK o una CLI
- Utilizzo ListUsers con un AWS SDK o una CLI
- Utilizzo ListVirtualMfaDevices con un AWS SDK o una CLI
- Utilizzo PutGroupPolicy con un AWS SDK o una CLI
- Utilizzo PutRolePermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutRolePolicy con un AWS SDK o una CLI
- Utilizzo PutUserPermissionsBoundary con un AWS SDK o una CLI
- Utilizzo PutUserPolicy con un AWS SDK o una CLI
- Utilizzo RemoveClientIdFromOpenIdConnectProvider con un AWS SDK o una CLI
- Utilizzo RemoveRoleFromInstanceProfile con un AWS SDK o una CLI
- Utilizzo RemoveUserFromGroup con un AWS SDK o una CLI
- Utilizzo ResyncMfaDevice con un AWS SDK o una CLI
- Utilizzo SetDefaultPolicyVersion con un AWS SDK o una CLI
- Utilizzo TagRole con un AWS SDK o una CLI
- Utilizzo TagUser con un AWS SDK o una CLI
- Utilizzo UntagRole con un AWS SDK o una CLI
- Utilizzo UntagUser con un AWS SDK o una CLI

- Utilizzo UpdateAccessKey con un AWS SDK o una CLI
- Utilizzo UpdateAccountPasswordPolicy con un AWS SDK o una CLI
- Utilizzo UpdateAssumeRolePolicy con un AWS SDK o una CLI
- Utilizzo UpdateGroup con un AWS SDK o una CLI
- Utilizzo UpdateLoginProfile con un AWS SDK o una CLI
- Utilizzo UpdateOpenIdConnectProviderThumbprint con un AWS SDK o una CLI
- Utilizzo UpdateRole con un AWS SDK o una CLI
- Utilizzo UpdateRoleDescription con un AWS SDK o una CLI
- Utilizzo UpdateSamlProvider con un AWS SDK o una CLI
- Utilizzo UpdateServerCertificate con un AWS SDK o una CLI
- Utilizzo UpdateSigningCertificate con un AWS SDK o una CLI
- Utilizzo UpdateUser con un AWS SDK o una CLI
- Utilizzo UploadServerCertificate con un AWS SDK o una CLI
- Utilizzo UploadSigningCertificate con un AWS SDK o una CLI

Utilizzo **`AddClientIdToOpenIdConnectProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AddClientIdToOpenIdConnectProvider`.

CLI

AWS CLI

Per aggiungere un ID client (audience) a un provider Open-ID Connect (OIDC)

Il `add-client-id-to-open-id-connect-provider` comando seguente aggiunge l'ID `my-application-ID` client al provider OIDC denominato. `server.example.com`

```
aws iam add-client-id-to-open-id-connect-provider \
    --client-id my-application-ID \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Questo comando non produce alcun output.

Per creare un provider OIDC, utilizzare il comando. `create-open-id-connect-provider`

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta Command AddClientIdToOpenIdConnectProviderReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando aggiunge l'ID client (o il pubblico) **`my-application-ID`** al provider OIDC esistente denominato. **`server.example.com`**

```
Add-IAMClientIDToOpenIDConnectProvider -ClientID "my-application-ID"
 -OpenIDConnectProviderARN "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
```

- Per i dettagli sull'API, vedere AddClientIdToOpenIdConnectProviderin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AddRoleToInstanceProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AddRoleToInstanceProfile`.

CLI

AWS CLI

Per aggiungere un ruolo a un profilo di istanza

Il `add-role-to-instance-profile` comando seguente aggiunge il ruolo denominato `S3Access` al profilo di istanza denominato`Webserver`.

```
aws iam add-role-to-instance-profile \
    --role-name S3Access \
```

```
    --instance-profile-name Webserver
```

Questo comando non produce alcun output.

Per creare un profilo di istanza, utilizzate il `create-instance-profile` comando.

Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta AddRoleToInstanceProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando aggiunge il ruolo denominato **`S3Access`** a un profilo di istanza esistente denominato**`webserver`**. Per creare il profilo dell'istanza, utilizzate il **`New-IAMInstanceProfile`** comando. Dopo aver creato il profilo dell'istanza e averlo associato a un ruolo utilizzando questo comando, puoi collegarlo a un'istanza EC2. A tale scopo, utilizza il **`New-EC2Instance`** cmdlet con il parametro **`InstanceProfile_Arn`** o con il **`InstanceProfile-Name`** parametro per avviare la nuova istanza.

```
Add-IAMRoleToInstanceProfile -RoleName "S3Access" -InstanceProfileName
 "webserver"
```

- Per i dettagli sull'API, vedere AddRoleToInstanceProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AddUserToGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AddUserToGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Add an existing IAM user to an existing IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to add.</param>
    /// <param name="groupName">The name of the group to add the user to.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AddUserToGroupAsync(string userName, string
groupName)
    {
        var response = await _IAMService.AddUserToGroupAsync(new
AddUserToGroupRequest
        {
            GroupName = groupName,
            UserName = userName,
        });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, AddUserToGroupconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Come aggiungere un utente a un gruppo IAM

Il comando `add-user-to-group` seguente aggiunte l'utente IAM denominato `Bob` al gruppo IAM denominato `Admins`.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di utenti in un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AddUserToGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando aggiunge l'utente denominato **Bob** al gruppo denominato**Admins**.

```
Add-IAMUserToGroup -UserName "Bob" -GroupName "Admins"
```

- Per i dettagli sull'API, vedere AddUserToGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AttachGroupPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AttachGroupPolicy`.

CLI

AWS CLI

Per allegare una policy gestita a un gruppo IAM

Il `attach-group-policy` comando seguente collega la policy AWS gestita denominata `ReadOnlyAccess` al gruppo IAM denominato`Finance`.

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachGroupPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio allega la policy gestita dai clienti denominata **`TesterPolicy`** al gruppo **`Testers`** IAM. Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nella versione predefinita di tale policy.

```
Register-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio allega la policy AWS gestita denominata **`AdministratorAccess`** al gruppo IAM. **`Admins`** Gli utenti di quel gruppo sono immediatamente interessati dalle autorizzazioni definite nell'ultima versione di tale policy.

```
Register-IAMGroupPolicy -GroupName Admins -PolicyArn arn:aws:iam::aws:policy/
AdministratorAccess
```

- Per i dettagli sull'API, vedere AttachGroupPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AttachRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AttachRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Gestione dei ruoli

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
```

```
  echo "  -p policy_ARN -- The IAM policy document ARN."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam attach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
  return 1
```

```
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::attachRolePolicy(const Aws::String &roleName,
                                   const Aws::String &policyArn,
                                   const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::ListAttachedRolePoliciesRequest list_request;
    list_request.SetRoleName(roleName);

    bool done = false;
    while (!done) {
        auto list_outcome = iam.ListAttachedRolePolicies(list_request);
        if (!list_outcome.IsSuccess()) {
            std::cerr << "Failed to list attached policies of role " <<
                      roleName << ": " << list_outcome.GetError().GetMessage() <<
                      std::endl;
            return false;
        }

        const auto &policies = list_outcome.GetResult().GetAttachedPolicies();
        if (std::any_of(policies.cbegin(), policies.cend(),
                        [=](const Aws::IAM::Model::AttachedPolicy &policy) {
```

```
                                return policy.GetPolicyArn() == policyArn;
                            })) {
            std::cout << "Policy " << policyArn <<
                      " is already attached to role " << roleName << std::endl;
            return true;
        }

        done = !list_outcome.GetResult().GetIsTruncated();
        list_request.SetMarker(list_outcome.GetResult().GetMarker());
    }

    Aws::IAM::Model::AttachRolePolicyRequest request;
    request.SetRoleName(roleName);
    request.SetPolicyArn(policyArn);

    Aws::IAM::Model::AttachRolePolicyOutcome outcome =
 iam.AttachRolePolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed to attach policy " << policyArn << " to role " <<
                  roleName << ": " << outcome.GetError().GetMessage() <<
 std::endl;
    }
    else {
        std::cout << "Successfully attached policy " << policyArn << " to role "
 <<
                  roleName << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come collegare una policy gestita a un ruolo IAM

Il `attach-role-policy` comando seguente collega la policy AWS gestita denominata `ReadOnlyAccess` al ruolo IAM denominato`ReadOnlyRole`.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(policyArn string, roleName string)
 error {
 _, err := wrapper.IamClient.AttachRolePolicy(context.TODO(),
 &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
```

```
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.AttachRolePolicyRequest;
import software.amazon.awssdk.services.iam.model.AttachedPolicy;
import software.amazon.awssdk.services.iam.model.ListAttachedRolePoliciesRequest;
import
 software.amazon.awssdk.services.iam.model.ListAttachedRolePoliciesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AttachRolePolicy {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                    <roleName> <policyArn>\s

                Where:
                    roleName - A role name that you can obtain from the AWS
Management Console.\s
                    policyArn - A policy ARN that you can obtain from the AWS
Management Console.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleName = args[0];
        String policyArn = args[1];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        attachIAMRolePolicy(iam, roleName, policyArn);
        iam.close();
    }

    public static void attachIAMRolePolicy(IamClient iam, String roleName, String
policyArn) {
        try {
            ListAttachedRolePoliciesRequest request =
ListAttachedRolePoliciesRequest.builder()
                    .roleName(roleName)
                    .build();

            ListAttachedRolePoliciesResponse response =
iam.listAttachedRolePolicies(request);
            List<AttachedPolicy> attachedPolicies = response.attachedPolicies();

            // Ensure that the policy is not attached to this role
            String polArn = "";
            for (AttachedPolicy policy : attachedPolicies) {
                polArn = policy.policyArn();
```

```
                if (polArn.compareTo(policyArn) == 0) {
                    System.out.println(roleName + " policy is already attached to
 this role.");
                    return;
                }
            }

            AttachRolePolicyRequest attachRequest =
 AttachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.attachRolePolicy(attachRequest);

            System.out.println("Successfully attached policy " + policyArn +
                    " to role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Collega la policy.

```
import { AttachRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";
```

```
const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 * @param {string} roleName
 */
export const attachRolePolicy = (policyArn, roleName) => {
  const command = new AttachRolePolicyCommand({
    PolicyArn: policyArn,
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var paramsRoleList = {
  RoleName: process.argv[2],
};
```

```
iam.listAttachedRolePolicies(paramsRoleList, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var myRolePolicies = data.AttachedPolicies;
    myRolePolicies.forEach(function (val, index, array) {
      if (myRolePolicies[index].PolicyName === "AmazonDynamoDBFullAccess") {
        console.log(
          "AmazonDynamoDBFullAccess is already attached to this role."
        );
        process.exit();
      }
    });
    var params = {
      PolicyArn: "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess",
      RoleName: process.argv[2],
    };
    iam.attachRolePolicy(params, function (err, data) {
      if (err) {
        console.log("Unable to attach policy to role", err);
      } else {
        console.log("Role attached successfully");
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun attachIAMRolePolicy(
    roleNameVal: String,
    policyArnVal: String
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }

        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkList(
    attachedPolicies: List<AttachedPolicy>,
    policyArnVal: String
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
```

```
        }
    }
    return 0
}
```

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
```

```
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

    public function attachRolePolicy($roleName, $policyArn)
    {
        return $this->customWaiter(function () use ($roleName, $policyArn) {
            $this->iamClient->attachRolePolicy([
                'PolicyArn' => $policyArn,
                'RoleName' => $roleName,
            ]);
        });
    }
```

- Per i dettagli sull'API, consulta la AttachRolePolicy in AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio collega la policy AWS gestita denominata **`SecurityAudit`** al ruolo **`CoSecurityAuditors`** IAM. Gli utenti che assumono quel ruolo sono immediatamente interessati dalle autorizzazioni definite nell'ultima versione di tale policy.

```
Register-IAMRolePolicy -RoleName CoSecurityAuditors -PolicyArn
 arn:aws:iam::aws:policy/SecurityAudit
```

- Per i dettagli sull'API, vedere AttachRolePolicy in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Collega una policy a un ruolo utilizzando l'oggetto Policy Boto3.

```
def attach_to_role(role_name, policy_arn):
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Policy(policy_arn).attach_role(RoleName=role_name)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise
```

Collega una policy a un ruolo utilizzando l'oggetto Role Boto3.

```
def attach_policy(role_name, policy_arn):
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise
```

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, allega e scollega le politiche relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end
```

```
 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end

 # Attaches a policy to a role
 #
 # @param role_name [String] The name of the role
 # @param policy_arn [String] The policy ARN
 # @return [Boolean] true if successful, false otherwise
 def attach_policy_to_role(role_name, policy_arn)
   @iam_client.attach_role_policy(
     role_name: role_name,
     policy_arn: policy_arn
   )
   true
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error attaching policy to role: #{e.message}")
   false
 end

 # Lists policy ARNs attached to a role
 #
 # @param role_name [String] The name of the role
 # @return [Array<String>] List of policy ARNs
 def list_attached_policy_arns(role_name)
   response = @iam_client.list_attached_role_policies(role_name: role_name)
   response.attached_policies.map(&:policy_arn)
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error listing policies attached to role: #{e.message}")
```

```
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sulle API, consulta AttachRolePolicy in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn attach_role_policy(
    client: &iamClient,
    role: &Role,
    policy: &Policy,
) -> Result<AttachRolePolicyOutput, SdkError<AttachRolePolicyError>> {
    client
        .attach_role_policy()
        .role_name(role.role_name())
        .policy_arn(policy.arn().unwrap_or_default())
```

```
        .send()
        .await
}
```

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func attachRolePolicy(role: String, policyArn: String) async throws {
        let input = AttachRolePolicyInput(
            policyArn: policyArn,
            roleName: role
        )
        do {
            _ = try await client.attachRolePolicy(input: input)
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta AttachRolePolicy in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AttachUserPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AttachUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

CLI

AWS CLI

Come collegare una policy gestita a un utente IAM

Il `attach-user-policy` comando seguente collega la policy AWS gestita denominata `AdministratorAccess` all'utente IAM denominato. `Alice`

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy gestite e policy inline nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AttachUserPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio AWS allega la policy gestita denominata **`AmazonCognitoPowerUser`** all'utente **`Bob`** IAM. L'utente è immediatamente interessato dalle autorizzazioni definite nell'ultima versione di tale policy.

```
Register-IAMUserPolicy -UserName Bob -PolicyArn arn:aws:iam::aws:policy/
AmazonCognitoPowerUser
```

- Per i dettagli sull'API, vedere AttachUserPolicy in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def attach_policy(user_name, policy_arn):
    """
    Attaches a policy to a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to user %s.", policy_arn,
 user_name)
        raise
```

- Per i dettagli sull'API, consulta AttachUserPolicy in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Attaches a policy to a user
#
# @param user_name [String] The name of the user
# @param policy_arn [String] The Amazon Resource Name (ARN) of the policy
# @return [Boolean] true if successful, false otherwise
def attach_policy_to_user(user_name, policy_arn)
  @iam_client.attach_user_policy(
    user_name: user_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error attaching policy to user: #{e.message}")
  false
end
```

- Per i dettagli sull'API, consulta la AttachUserPolicy in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn attach_user_policy(
```

```
    client: &iamClient,
    user_name: &str,
    policy_arn: &str,
) -> Result<(), iamError> {
    client
        .attach_user_policy()
        .user_name(user_name)
        .policy_arn(policy_arn)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sull'API, consulta AttachUserPolicy in AWS SDK for Rust API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ChangePassword`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ChangePassword`.

CLI

AWS CLI

Per modificare la password del tuo utente IAM

Per modificare la password del tuo utente IAM, ti consigliamo di utilizzare il `--cli-input-json` parametro per passare un file JSON che contenga la vecchia e la nuova password. Utilizzando questo metodo, puoi utilizzare password complesse con caratteri non alfanumerici. Può essere difficile utilizzare password con caratteri non alfanumerici quando le si passano come parametri della riga di comando. Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `change-password` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam change-password \
    --generate-cli-skeleton > change-password.json
```

Il comando precedente crea un file JSON chiamato change-password.json che potete utilizzare per inserire la vecchia e la nuova password. Ad esempio, il file potrebbe avere l'aspetto seguente.

```
{
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

Quindi, per modificare la password, usa nuovamente il `change-password` comando, questa volta passando il `--cli-input-json` parametro per specificare il file JSON. Il `change-password` comando seguente utilizza il `--cli-input-json` parametro con un file JSON chiamato change-password.json.

```
aws iam change-password \
    --cli-input-json file://change-password.json
```

Questo comando non produce alcun output.

Questo comando può essere chiamato solo dagli utenti IAM. Se questo comando viene chiamato utilizzando le credenziali AWS dell'account (root), restituisce un `InvalidUserType` errore.

Per ulteriori informazioni, consulta Come un utente IAM modifica la propria password nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta ChangePassword AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando modifica la password dell'utente che esegue il comando. Questo comando può essere chiamato solo dagli utenti IAM. Se questo comando viene chiamato quando si accede con le credenziali AWS dell'account (root), restituisce un errore. **`InvalidUserType`**

```
Edit-IAMPassword -OldPassword "MyOldP@ssw0rd" -NewPassword "MyNewP@ssw0rd"
```

- Per i dettagli sull'API, vedere ChangePasswordin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateAccessKey`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle chiavi di accesso

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
CreateAccessKeyRequest
```

```
        {
            UserName = userName,
        });

        return response.AccessKey;

    }
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
```

```
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
Aws::String AwsDoc::IAM::createAccessKey(const Aws::String &userName,
                                         const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
```

```
    Aws::IAM::Model::CreateAccessKeyRequest request;
    request.SetUserName(userName);

    Aws::String result;
    Aws::IAM::Model::CreateAccessKeyOutcome outcome =
 iam.CreateAccessKey(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating access key for IAM user " << userName
                  << ":" << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const auto &accessKey = outcome.GetResult().GetAccessKey();
        std::cout << "Successfully created access key for IAM user " <<
                  userName << std::endl << "  aws_access_key_id = " <<
                  accessKey.GetAccessKeyId() << std::endl <<
                  " aws_secret_access_key = " << accessKey.GetSecretAccessKey()
 <<
                  std::endl;
        result = accessKey.GetAccessKeyId();
    }

    return result;
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come creare una chiave di accesso per un utente IAM

Il comando `create-access-key` seguente crea una chiave di accesso (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam create-access-key \
    --user-name Bob
```

Output:

```
{
```

```
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

Conserva la chiave di accesso segreta in un luogo sicuro. Se viene persa, non può essere recuperata e dovrai creare una nuova chiave di accesso.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta CreateAccessKey in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
func (wrapper UserWrapper) CreateAccessKeyPair(userName string)
 (*types.AccessKey, error) {
 var key *types.AccessKey
```

```
 result, err := wrapper.IamClient.CreateAccessKey(context.TODO(),
 &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.CreateAccessKeyRequest;
import software.amazon.awssdk.services.iam.model.CreateAccessKeyResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateAccessKey {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                   <user>\s

                Where:
                   user - An AWS IAM user that you can obtain from the AWS
 Management Console.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String user = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String keyId = createIAMAccessKey(iam, user);
        System.out.println("The Key Id is " + keyId);
        iam.close();
    }

    public static String createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();

            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey().accessKeyId();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea la chiave di accesso.

```
import { CreateAccessKeyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 */
export const createAccessKey = (userName) => {
  const command = new CreateAccessKeyCommand({ UserName: userName });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.createAccessKey({ UserName: "IAM_USER_NAME" }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.AccessKey);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createIAMAccessKey(user: String?): String {
    val request =
        CreateAccessKeyRequest {
            userName = user
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createAccessKey(request)
        return response.accessKey?.accessKeyId.toString()
```

```
    }
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea una nuova chiave di accesso e una coppia di chiavi di accesso segrete e le assegna all'utente**David**. Assicurati di salvare i **SecretAccessKey** valori **AccessKeyId** and in un file perché è l'unica volta che puoi ottenere il**SecretAccessKey**. Non puoi recuperarla in un secondo momento. Se si perde la chiave segreta, è necessario creare una nuova coppia di chiavi di accesso.

```
New-IAMAccessKey -UserName David
```

Output:

```
AccessKeyId     : AKIAIOSFODNN7EXAMPLE
CreateDate      : 4/13/2015 1:00:42 PM
SecretAccessKey : wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Status          : Active
UserName        : David
```

- Per i dettagli sull'API, vedere CreateAccessKey in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
        raise
    else:
        return key_pair
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
```

```
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity => e
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded => e
    @logger.error("Error creating access key: limit exceeded. Cannot create
more.")
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
```

```
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: "Inactive"
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn create_access_key(client: &iamClient, user_name: &str) ->
 Result<AccessKey, iamError> {
    let mut tries: i32 = 0;
    let max_tries: i32 = 10;

    let response: Result<CreateAccessKeyOutput, SdkError<CreateAccessKeyError>> =
 loop {
        match client.create_access_key().user_name(user_name).send().await {
            Ok(inner_response) => {
                break Ok(inner_response);
            }
            Err(e) => {
                tries += 1;
                if tries > max_tries {
                    break Err(e);
                }
                sleep(Duration::from_secs(2)).await;
            }
        }
    };

    Ok(response.unwrap().access_key.unwrap())
}
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public func createAccessKey(userName: String) async throws ->
IAMClientTypes.AccessKey {
       let input = CreateAccessKeyInput(
           userName: userName
       )
       do {
           let output = try await iamClient.createAccessKey(input: input)
           guard let accessKey = output.accessKey else {
               throw ServiceHandlerError.keyError
           }
           return accessKey
       } catch {
           throw error
       }
   }
```

- Per i dettagli sull'API, consulta CreateAccessKey in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateAccountAlias`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateAccountAlias`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::createAccountAlias(const Aws::String &aliasName,
                                     const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::CreateAccountAliasRequest request;
    request.SetAccountAlias(aliasName);

    Aws::IAM::Model::CreateAccountAliasOutcome outcome = iam.CreateAccountAlias(
            request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating account alias " << aliasName << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully created account alias " << aliasName <<
                  std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come Creare l'alias di un account

Il `create-account-alias` comando seguente crea l'alias `examplecorp` per il tuo AWS account.

```
aws iam create-account-alias \
    --account-alias examplecorp
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Your AWS account ID and its alias nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.CreateAccountAliasRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateAccountAlias {
    public static void main(String[] args) {
        final String usage = """
                Usage:
                    <alias>\s

                Where:
                    alias - The account alias to create (for example,
 myawsaccount).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alias = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        createIAMAccountAlias(iam, alias);
        iam.close();
        System.out.println("Done");
    }

    public static void createIAMAccountAlias(IamClient iam, String alias) {
        try {
            CreateAccountAliasRequest request =
 CreateAccountAliasRequest.builder()
                    .accountAlias(alias)
                    .build();

            iam.createAccountAlias(request);
            System.out.println("Successfully created account alias: " + alias);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }
}
```

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea l'alias dell'account.

```
import { CreateAccountAliasCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} alias - A unique name for the account alias.
 * @returns
 */
export const createAccountAlias = (alias) => {
  const command = new CreateAccountAliasCommand({
    AccountAlias: alias,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.createAccountAlias({ AccountAlias: process.argv[2] }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createIAMAccountAlias(alias: String) {
    val request =
        CreateAccountAliasRequest {
            accountAlias = alias
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.createAccountAlias(request)
        println("Successfully created account alias named $alias")
    }
}
```

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio modifica l'alias AWS dell'account in**`mycompanyaws`**. L'indirizzo della pagina di accesso dell'utente cambia in https://mycompanyaws.signin.aws.amazon.com/console. L'URL originale che utilizza il numero ID dell'account anziché l'alias (https://<accountidnumber>.signin.aws.amazon.com/console) continua a funzionare. Tuttavia, tutti gli URL basati su alias precedentemente definiti smettono di funzionare.

```
New-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, vedere CreateAccountAlias in Cmdlet Reference.AWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_alias(alias):
    """
    Creates an alias for the current account. The alias can be used in place of
 the
    account ID in the sign-in URL. An account can have only one alias. When a new
    alias is created, it replaces any existing alias.

    :param alias: The alias to assign to the account.
    """

    try:
        iam.create_account_alias(AccountAlias=alias)
        logger.info("Created an alias '%s' for your account.", alias)
    except ClientError:
        logger.exception("Couldn't create alias '%s' for your account.", alias)
        raise
```

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
```

```
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info("Account aliases are:")
      response.account_aliases.each { |account_alias| @logger.info("
 #{account_alias}") }
    else
      @logger.info("No account aliases found.")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta CreateAccountAlias in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM group.
    /// </summary>
    /// <param name="groupName">The name to give the IAM group.</param>
    /// <returns>The IAM group that was created.</returns>
    public async Task<Group> CreateGroupAsync(string groupName)
    {
        var response = await _IAMService.CreateGroupAsync(new CreateGroupRequest
{ GroupName = groupName });
        return response.Group;
    }
```

- Per i dettagli sull'API, consulta la CreateGroupsezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Come creare un gruppo IAM

Il comando `create-group` seguente crea un gruppo IAM denominato `Admins`.

```
aws iam create-group \
    --group-name Admins
```

Output:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-03-09T20:30:24.940Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

Per ulteriori informazioni, consulta Creazione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreateGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { CreateGroupCommand, IAMClient } from "@aws-sdk/client-iam";
```

```
const client = new IAMClient({});

/**
 *
 * @param {string} groupName
 */
export const createGroup = async (groupName) => {
  const command = new CreateGroupCommand({ GroupName: groupName });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per i dettagli sull'API, consulta la CreateGroupsezione AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea un nuovo gruppo IAM denominato**`Developers`**.

```
New-IAMGroup -GroupName Developers
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 4/14/2015 11:21:31 AM
GroupId    : QNEJ5PM4NFSQCEXAMPLE1
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere CreateGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateInstanceProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione e gestione di un servizio resiliente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Create a policy, role, and profile that is associated with instances with
a specified name.
   /// An instance's associated profile defines a role that is assumed by the
   /// instance.The role has attached policies that specify the AWS permissions
granted to
   /// clients that run on the instance.
   /// </summary>
   /// <param name="policyName">Name to use for the policy.</param>
   /// <param name="roleName">Name to use for the role.</param>
   /// <param name="profileName">Name to use for the profile.</param>
   /// <param name="ssmOnlyPolicyFile">Path to a policy file for SSM.</param>
   /// <param name="awsManagedPolicies">AWS Managed policies to be attached to
the role.</param>
   /// <returns>The Arn of the profile.</returns>
   public async Task<string> CreateInstanceProfileWithName(
       string policyName,
       string roleName,
       string profileName,
       string ssmOnlyPolicyFile,
       List<string>? awsManagedPolicies = null)
   {
```

```
        var assumeRoleDoc = "{" +
                                    "\"Version\": \"2012-10-17\"," +
                                    "\"Statement\": [{" +
                                         "\"Effect\": \"Allow\"," +
                                         "\"Principal\": {" +
                                         "\"Service\": [" +
                                             "\"ec2.amazonaws.com\"" +
                                         "]" +
                                         "}," +
                                    "\"Action\": \"sts:AssumeRole\"" +
                                    "}]" +
                                "}";

        var policyDocument = await File.ReadAllTextAsync(ssmOnlyPolicyFile);

        var policyArn = "";

        try
        {
            var createPolicyResult = await _amazonIam.CreatePolicyAsync(
                new CreatePolicyRequest
                {
                    PolicyName = policyName,
                    PolicyDocument = policyDocument
                });
            policyArn = createPolicyResult.Policy.Arn;
        }
        catch (EntityAlreadyExistsException)
        {
            // The policy already exists, so we look it up to get the Arn.
            var policiesPaginator = _amazonIam.Paginators.ListPolicies(
                new ListPoliciesRequest()
                {
                    Scope = PolicyScopeType.Local
                });
            // Get the entire list using the paginator.
            await foreach (var policy in policiesPaginator.Policies)
            {
                if (policy.PolicyName.Equals(policyName))
                {
                    policyArn = policy.Arn;
                }
            }
```

```
            if (policyArn == null)
            {
                throw new InvalidOperationException("Policy not found");
            }
        }

        try
        {
            await _amazonIam.CreateRoleAsync(new CreateRoleRequest()
            {
                RoleName = roleName,
                AssumeRolePolicyDocument = assumeRoleDoc,
            });
            await _amazonIam.AttachRolePolicyAsync(new AttachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = policyArn
            });
            if (awsManagedPolicies != null)
            {
                foreach (var awsPolicy in awsManagedPolicies)
                {
                    await _amazonIam.AttachRolePolicyAsync(new
AttachRolePolicyRequest()
                    {
                        PolicyArn = $"arn:aws:iam::aws:policy/{awsPolicy}",
                        RoleName = roleName
                    });
                }
            }
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Role already exists.");
        }

        string profileArn = "";
        try
        {
            var profileCreateResponse = await
_amazonIam.CreateInstanceProfileAsync(
                new CreateInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
```

```
                });
            // Allow time for the profile to be ready.
            profileArn = profileCreateResponse.InstanceProfile.Arn;
            Thread.Sleep(10000);
            await _amazonIam.AddRoleToInstanceProfileAsync(
                new AddRoleToInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });

        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Policy already exists.");
            var profileGetResponse = await _amazonIam.GetInstanceProfileAsync(
                new GetInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            profileArn = profileGetResponse.InstanceProfile.Arn;
        }
        return profileArn;
    }
```

- Per i dettagli sull'API, consulta CreateInstanceProfile in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come creare un profilo dell'istanza

Il comando `create-instance-profile` seguente crea un profilo dell'istanza denominato `Webserver`.

```
aws iam create-instance-profile \
    --instance-profile-name Webserver
```

Output:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJMBYC7DLSPEXAMPLE",
        "Roles": [],
        "CreateDate": "2015-03-09T20:33:19.626Z",
        "InstanceProfileName": "Webserver",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
    }
}
```

Per aggiungere un ruolo a un profilo dell'istanza, usa il comando `add-role-to-instance-profile`.

Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreateInstanceProfile in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const { InstanceProfile } = await iamClient.send(
  new CreateInstanceProfileCommand({
    InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
  }),
);
await waitUntilInstanceProfileExists(
  { client: iamClient },
  { InstanceProfileName: NAMES.ssmOnlyInstanceProfileName },
);
```

- Per i dettagli sull'API, consulta CreateInstanceProfile in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea un nuovo profilo di istanza IAM denominato**`ProfileForDevEC2Instance`**. È necessario eseguire il **`Add-IAMRoleToInstanceProfile`** comando separatamente per associare il profilo dell'istanza a un ruolo IAM esistente che fornisce le autorizzazioni all'istanza. Infine, collega il profilo dell'istanza a un'istanza EC2 quando la avvii. A tale scopo, utilizza il **`New-EC2Instance`** cmdlet con il **`InstanceProfile_Arn`** parametro or. **`InstanceProfile_Name`**

```
New-IAMInstanceProfile -InstanceProfileName ProfileForDevEC2Instance
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/
ProfileForDevEC2Instance
CreateDate          : 4/14/2015 11:31:39 AM
InstanceProfileId   : DYMFXL556EY46EXAMPLE1
InstanceProfileName : ProfileForDevEC2Instance
Path                : /
Roles               : {}
```

- Per i dettagli sull'API, vedere CreateInstanceProfile in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo esempio crea una policy, un ruolo e un profilo dell'istanza e li collega tutti insieme.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
 created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
 that is
                          created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
        self.key_pair_name = f"{resource_prefix}-key-pair"
```

```
   def create_instance_profile(
       self, policy_file, policy_name, role_name, profile_name,
aws_managed_policies=()
   ):
       """
       Creates a policy, role, and profile that is associated with instances
created by
       this class. An instance's associated profile defines a role that is
assumed by the
       instance. The role has attached policies that specify the AWS permissions
granted to
       clients that run on the instance.

       :param policy_file: The name of a JSON file that contains the policy
definition to
                           create and attach to the role.
       :param policy_name: The name to give the created policy.
       :param role_name: The name to give the created role.
       :param profile_name: The name to the created profile.
       :param aws_managed_policies: Additional AWS-managed policies that are
attached to
                                    the role, such as
AmazonSSMManagedInstanceCore to grant
                                    use of Systems Manager to send commands to
the instance.
       :return: The ARN of the profile that is created.
       """
       assume_role_doc = {
           "Version": "2012-10-17",
           "Statement": [
               {
                   "Effect": "Allow",
                   "Principal": {"Service": "ec2.amazonaws.com"},
                   "Action": "sts:AssumeRole",
               }
           ],
       }
       with open(policy_file) as file:
           instance_policy_doc = file.read()

       policy_arn = None
       try:
```

```
            pol_response = self.iam_client.create_policy(
                PolicyName=policy_name, PolicyDocument=instance_policy_doc
            )
            policy_arn = pol_response["Policy"]["Arn"]
            log.info("Created policy with ARN %s.", policy_arn)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                log.info("Policy %s already exists, nothing to do.", policy_name)
                list_pol_response = self.iam_client.list_policies(Scope="Local")
                for pol in list_pol_response["Policies"]:
                    if pol["PolicyName"] == policy_name:
                        policy_arn = pol["Arn"]
                        break
            if policy_arn is None:
                raise AutoScalerError(f"Couldn't create policy {policy_name}:
{err}")

        try:
            self.iam_client.create_role(
                RoleName=role_name,
AssumeRolePolicyDocument=json.dumps(assume_role_doc)
            )
            self.iam_client.attach_role_policy(RoleName=role_name,
PolicyArn=policy_arn)
            for aws_policy in aws_managed_policies:
                self.iam_client.attach_role_policy(
                    RoleName=role_name,
                    PolicyArn=f"arn:aws:iam::aws:policy/{aws_policy}",
                )
            log.info("Created role %s and attached policy %s.", role_name,
policy_arn)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                log.info("Role %s already exists, nothing to do.", role_name)
            else:
                raise AutoScalerError(f"Couldn't create role {role_name}: {err}")

        try:
            profile_response = self.iam_client.create_instance_profile(
                InstanceProfileName=profile_name
            )
            waiter = self.iam_client.get_waiter("instance_profile_exists")
            waiter.wait(InstanceProfileName=profile_name)
            time.sleep(10)  # wait a little longer
```

```
            profile_arn = profile_response["InstanceProfile"]["Arn"]
            self.iam_client.add_role_to_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )
            log.info("Created profile %s and added role %s.", profile_name,
role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                prof_response = self.iam_client.get_instance_profile(
                    InstanceProfileName=profile_name
                )
                profile_arn = prof_response["InstanceProfile"]["Arn"]
                log.info(
                    "Instance profile %s already exists, nothing to do.",
profile_name
                )
            else:
                raise AutoScalerError(
                    f"Couldn't create profile {profile_name} and attach it to
role\n"
                    f"{role_name}: {err}"
                )
        return profile_arn
```

- Per i dettagli sull'API, consulta CreateInstanceProfile in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateLoginProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLoginProfile`.

CLI

AWS CLI

Per creare una password per un utente IAM

Per creare una password per un utente IAM, consigliamo di utilizzare il `--cli-input-json` parametro per passare un file JSON che contenga la password. Utilizzando questo metodo, puoi creare una password sicura con caratteri non alfanumerici. Può essere difficile creare una password con caratteri non alfanumerici quando la si passa come parametro della riga di comando.

Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `create-login-profile` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam create-login-profile \
    --generate-cli-skeleton > create-login-profile.json
```

Il comando precedente crea un file JSON chiamato create-login-profile .json che potete utilizzare per inserire le informazioni per un comando successivo. `create-login-profile` Per esempio:

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

Successivamente, per creare una password per un utente IAM, utilizza nuovamente il `create-login-profile` comando, questa volta passando il `--cli-input-json` parametro per specificare il file JSON. Il `create-login-profile` comando seguente utilizza il `--cli-input-json` parametro con un file JSON chiamato create-login-profile .json.

```
aws iam create-login-profile \
    --cli-input-json file://create-login-profile.json
```

Output:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2015-03-10T20:55:40.274Z",
        "PasswordResetRequired": true
    }
```

```
}
```

Se la nuova password viola la politica relativa alle password dell'account, il comando restituisce un errore. `PasswordPolicyViolation`

Per modificare la password di un utente che ne ha già una, usa`update-login-profile`. Per impostare una politica di password per l'account, usa il `update-account-password-policy` comando.

Se la politica sulla password dell'account lo consente, gli utenti IAM possono modificare le proprie password utilizzando il `change-password` comando.

Per ulteriori informazioni, consulta la sezione Gestione delle password per gli utenti IAM nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta CreateLoginProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea una password (temporanea) per l'utente IAM di nome Bob e imposta il flag che richiede all'utente di modificare la password al **Bob** successivo accesso.

```
New-IAMLoginProfile -UserName Bob -Password P@ssw0rd -PasswordResetRequired $true
```

Output:

```
CreateDate                 PasswordResetRequired             UserName
----------                 ---------------------             --------
4/14/2015 12:26:30 PM      True                              Bob
```

- Per i dettagli sull'API, vedere CreateLoginProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateOpenIdConnectProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateOpenIdConnectProvider`.

CLI

AWS CLI

Per creare un provider OpenID Connect (OIDC)

Per creare un provider OpenID Connect (OIDC), consigliamo di utilizzare il `--cli-input-json` parametro per passare un file JSON contenente i parametri richiesti. Quando si crea un provider OIDC, è necessario passare l'URL del provider e l'URL deve iniziare con. `https://` Può essere difficile passare l'URL come parametro della riga di comando, perché i caratteri due punti (:) e barra (/) hanno un significato speciale in alcuni ambienti della riga di comando. L'utilizzo del `--cli-input-json` parametro consente di aggirare questa limitazione.

Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `create-open-id-connect-provider` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam create-open-id-connect-provider \
    --generate-cli-skeleton > create-open-id-connect-provider.json
```

Il comando precedente crea un file JSON chiamato create-open-id-connect -provider.json che potete utilizzare per inserire le informazioni per un comando successivo. `create-open-id-connect-provider` Per esempio:

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
    ]
}
```

Successivamente, per creare il provider OpenID Connect (OIDC), utilizzate nuovamente il `create-open-id-connect-provider` comando, questa volta passando il `--cli-input-json` parametro per specificare il file JSON. Il `create-open-id-connect-provider`

comando seguente utilizza il `--cli-input-json` parametro con un file JSON chiamato - provider.json. create-open-id-connect

```
aws iam create-open-id-connect-provider \
    --cli-input-json file://create-open-id-connect-provider.json
```

Output:

```
{
    "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Per ulteriori informazioni sui provider OIDC, consulta Creating OpenID Connect (OIDC) Identity Provider nella IAM User Guide.AWS

Per ulteriori informazioni su come ottenere impronte digitali per un provider OIDC, consulta Ottenere l'impronta personale per un provider di identità OpenID Connect nella IAM User Guide.AWS

- Per i dettagli sull'API, consulta Provider in Command Reference. CreateOpen IdConnect AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea un provider IAM OIDC associato al servizio provider compatibile con OIDC che si trova nell'URL **`https://example.oidcprovider.com`** e nell'ID client. **`my-testapp-1`** Il provider OIDC fornisce l'impronta digitale. Per autenticare l'impronta personale, segui la procedura all'indirizzo http://docs.aws.amazon.com/IAM/latest/ / identity-providers-oidc-obtain-thumbprint.html. UserGuide

```
New-IAMOpenIDConnectProvider -Url https://example.oidcprovider.com -ClientIDList
 my-testapp-1 -ThumbprintList 990F419EXAMPLEECF12DDEDA5EXAMPLE52F20D9E
```

Output:

```
arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, CreateOpenIdConnectvedere Provider in Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreatePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreatePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle policy
- Lavora con l'API IAM Policy Builder

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
```

```
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });

        return response.Policy;
    }
```

- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
```

```
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
```

```
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}
```

- Per i dettagli sull'API, consulta CreatePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
Aws::String AwsDoc::IAM::createPolicy(const Aws::String &policyName,
                                      const Aws::String &rsrcArn,
                                      const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::CreatePolicyRequest request;
    request.SetPolicyName(policyName);
```

```
    request.SetPolicyDocument(BuildSamplePolicyDocument(rsrcArn));

    Aws::IAM::Model::CreatePolicyOutcome outcome = iam.CreatePolicy(request);
    Aws::String result;
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating policy " << policyName << ": " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        result = outcome.GetResult().GetPolicy().GetArn();
        std::cout << "Successfully created policy " << policyName <<
                  std::endl;
    }

    return result;
}

Aws::String AwsDoc::IAM::BuildSamplePolicyDocument(const Aws::String &rsrc_arn) {
    std::stringstream stringStream;
    stringStream << "{"
                 << "  \"Version\": \"2012-10-17\","
                 << "  \"Statement\": ["
                 << "    {"
                 << "        \"Effect\": \"Allow\","
                 << "        \"Action\": \"logs:CreateLogGroup\","
                 << "        \"Resource\": \""
                 << rsrc_arn
                 << "\""
                 << "    },"
                 << "    {"
                 << "        \"Effect\": \"Allow\","
                 << "        \"Action\": ["
                 << "            \"dynamodb:DeleteItem\","
                 << "            \"dynamodb:GetItem\","
                 << "            \"dynamodb:PutItem\","
                 << "            \"dynamodb:Scan\","
                 << "            \"dynamodb:UpdateItem\""
                 << "       ],"
                 << "       \"Resource\": \""
                 << rsrc_arn
                 << "\""
                 << "    }"
                 << "   ]"
                 << "}";
```

```
    return stringStream.str();
}
```

- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare una policy gestita dal cliente

Il comando seguente crea una policy gestita dal cliente denominata `my-policy`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy
```

Il file `policy` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura alla cartella `shared` in un bucket Amazon S3 denominato `my-bucket`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/shared/*"
            ]
        }
    ]
}
```

Output:

```
{
```

```
    "Policy": {
        "PolicyName": "my-policy",
        "CreateDate": "2015-06-01T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
        "UpdateDate": "2015-06-01T19:31:18.620Z"
    }
}
```

Per ulteriori informazioni sull'utilizzo dei file come input per i parametri di stringa, [consultate Specificare i valori dei parametri per la AWS CLI nella AWS CLI](#) User Guide.

Esempio 2: Come creare una policy gestita dal cliente con una descrizione

Il comando seguente crea una policy gestita dal cliente denominata `my-policy` con una descrizione non modificabile.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --description "This policy grants access to all Put, Get, and List actions
 for my-bucket"
```

Il file `policy.json` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura a tutte le operazioni Put, List e Get per un bucket Amazon S3 denominato `my-bucket`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                 "s3:ListBucket*",
                 "s3:PutBucket*",
                 "s3:GetBucket*"
             ],
             "Resource": [
```

```
                "arn:aws:s3:::my-bucket"
            ]
        }
    ]
}
```

Output:

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T22:38:47+00:00",
        "UpdateDate": "2023-05-24T22:38:47+00:00"
    }
}
```

Per ulteriori informazioni sulle policy basate sull'identità consulta Policy basate sulle identità e policy basate su risorse nella Guida per l'utente IAM AWS .

Esempio 3: Come creare una policy gestita dal cliente con tag

Il comando seguente crea una policy gestita dal cliente denominata `my-policy` con tag. Questo esempio utilizza il flag del parametro `--tags` con i seguenti tag in formato JSON: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Il file `policy.json` è un documento JSON nella cartella corrente che consente l'accesso in sola lettura a tutte le operazioni Put, List e Get per un bucket Amazon S3 denominato `my-bucket`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                 "s3:ListBucket*",
                 "s3:PutBucket*",
                 "s3:GetBucket*"
             ],
             "Resource": [
                 "arn:aws:s3:::my-bucket"
             ]
         }
     ]
 }
```

Output:

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "ANPAWGSUGIDPEXAMPLE",
        "Arn": "arn:aws:iam::12345678012:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-05-24T23:16:39+00:00",
        "UpdateDate": "2023-05-24T23:16:39+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
                "Key": "Location",
                "Value": "Seattle"
            {
```

```
        ]
    }
}
```

Per ulteriori informazioni sulle policy di applicazione di tag, consulta Applicazione di tag a policy gestite dal cliente nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command CreatePolicyReference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(policyName string, actions []string,
  resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
```

```
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(resourceArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(context.TODO(),
 &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}
```

- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.services.iam.model.CreatePolicyRequest;
import software.amazon.awssdk.services.iam.model.CreatePolicyResponse;
import software.amazon.awssdk.services.iam.model.GetPolicyRequest;
```

```
import software.amazon.awssdk.services.iam.model.GetPolicyResponse;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.waiters.IamWaiter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreatePolicy {

    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"dynamodb:DeleteItem\"," +
            "            \"dynamodb:GetItem\"," +
            "            \"dynamodb:PutItem\"," +
            "            \"dynamodb:Scan\"," +
            "            \"dynamodb:UpdateItem\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static void main(String[] args) {

        final String usage = """
                Usage:
                    CreatePolicy <policyName>\s

                Where:
                    policyName - A unique policy name.\s
                """;
```

```
        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String policyName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMPolicy(iam, policyName);
        System.out.println("Successfully created a policy with this ARN value: "
+ result);
        iam.close();
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();

            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument)
                    .build();

            CreatePolicyResponse response = iam.createPolicy(request);

            // Wait until the policy is created.
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();

            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
iamWaiter.waitUntilPolicyExists(polRequest);

waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
            return response.policy().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
        return "";
    }
}
```

- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea la policy.

```
import { CreatePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyName
 */
export const createPolicy = (policyName) => {
  const command = new CreatePolicyCommand({
    PolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Action: "*",
          Resource: "*",
        },
      ],
    }),
    PolicyName: policyName,
  });
```

```
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var myManagedPolicy = {
  Version: "2012-10-17",
  Statement: [
    {
      Effect: "Allow",
      Action: "logs:CreateLogGroup",
      Resource: "RESOURCE_ARN",
    },
    {
      Effect: "Allow",
      Action: [
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Scan",
        "dynamodb:UpdateItem",
      ],
```

```
      Resource: "RESOURCE_ARN",
    },
  ],
};

var params = {
  PolicyDocument: JSON.stringify(myManagedPolicy),
  PolicyName: "myDynamoDBPolicy",
};

iam.createPolicy(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createIAMPolicy(policyNameVal: String?): String {
    val policyDocumentVal =
        "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
```

```
            "            \"dynamodb:DeleteItem\"," +
            "            \"dynamodb:GetItem\"," +
            "            \"dynamodb:PutItem\"," +
            "            \"dynamodb:Scan\"," +
            "            \"dynamodb:UpdateItem\"" +
            "        ]," +
            "        \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}"

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}
```

- Per i dettagli sull'API, CreatePolicyconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
```

```
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

    public function createPolicy(string $policyName, string $policyDocument)
    {
        $result = $this->customWaiter(function () use ($policyName,
 $policyDocument) {
            return $this->iamClient->createPolicy([
                'PolicyName' => $policyName,
                'PolicyDocument' => $policyDocument,
            ]);
        });
        return $result['Policy'];
    }
```

- Per i dettagli sull'API, consulta la CreatePolicysezione AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea una nuova policy IAM nell' AWS account corrente denominato **MySamplePolicy** Il file **MySamplePolicy.json** fornisce il contenuto della policy. Si noti che è necessario utilizzare il parametro **-Raw** switch per elaborare correttamente il file di policy JSON.

```
New-IAMPolicy -PolicyName MySamplePolicy -PolicyDocument (Get-Content -Raw
 MySamplePolicy.json)
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 4/14/2015 2:45:59 PM
DefaultVersionId : v1
Description      :
```

```
IsAttachable     : True
Path             : /
PolicyId         : LD4KP6HVFE7WGEXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 4/14/2015 2:45:59 PM
```

- Per i dettagli sull'API, vedere CreatePolicyin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::my-bucket/*' to allow actions on all
 objects
                         in the bucket named 'my-bucket'.
    :return: The newly created policy.
    """
    policy_doc = {
        "Version": "2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
```

```
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy
```

- Per i dettagli sull'API, consulta CreatePolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, allega e scollega le politiche relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
```

```
 # @return [String] The policy ARN if successful, otherwise nil
 def create_policy(policy_name, policy_document)
   response = @iam_client.create_policy(
     policy_name: policy_name,
     policy_document: policy_document.to_json
   )
   response.policy.arn
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error creating policy: #{e.message}")
   nil
 end

 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end

 # Attaches a policy to a role
 #
 # @param role_name [String] The name of the role
 # @param policy_arn [String] The policy ARN
 # @return [Boolean] true if successful, false otherwise
 def attach_policy_to_role(role_name, policy_arn)
   @iam_client.attach_role_policy(
     role_name: role_name,
     policy_arn: policy_arn
   )
   true
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error attaching policy to role: #{e.message}")
```

```
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sulle API, consulta la sezione AWS SDK for Ruby API CreatePolicyReference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn create_policy(
    client: &iamClient,
    policy_name: &str,
    policy_document: &str,
) -> Result<Policy, iamError> {
    let policy = client
        .create_policy()
        .policy_name(policy_name)
        .policy_document(policy_document)
        .send()
        .await?;
    Ok(policy.policy.unwrap())
}
```

- Per i dettagli sulle API, consulta la CreatePolicyguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func createPolicy(name: String, policyDocument: String) async throws -
> IAMClientTypes.Policy {
        let input = CreatePolicyInput(
            policyDocument: policyDocument,
            policyName: name
        )
        do {
            let output = try await iamClient.createPolicy(input: input)
            guard let policy = output.policy else {
                throw ServiceHandlerError.noSuchPolicy
            }
            return policy
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreatePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreatePolicyVersion`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreatePolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle policy

CLI

AWS CLI

Per creare una nuova versione di una policy gestita

Questo esempio crea una nuova versione v2 della policy IAM il cui ARN è `arn:aws:iam::123456789012:policy/MyPolicy` e la rende la versione predefinita.

```
aws iam create-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --policy-document file://NewPolicyVersion.json \
    --set-as-default
```

Output:

```
{
    "PolicyVersion": {
        "CreateDate": "2015-06-16T18:56:03.721Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

Per ulteriori informazioni, consulta Controllo delle versioni delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta CreatePolicyVersion in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea una nuova versione «v2" della policy IAM il cui ARN è **`arn:aws:iam::123456789012:policy/MyPolicy`** e la rende la versione predefinita. Il **`NewPolicyVersion.json`** file fornisce il contenuto della policy. Si noti che è necessario utilizzare il parametro **`-Raw`** switch per elaborare correttamente il file di policy JSON.

```
New-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy -
PolicyDocument (Get-content -Raw NewPolicyVersion.json) -SetAsDefault $true
```

Output:

```
CreateDate                          Document                  IsDefaultVersion
           VersionId
----------                          --------                  ---------------
           ---------
4/15/2015 10:54:54 AM                                         True
           v2
```

- Per i dettagli sull'API, vedere CreatePolicyVersion in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_policy_version(policy_arn, actions, resource_arn, set_as_default):
    """
    Creates a policy version. Policies can have up to five versions. The default
    version is the one that is used for all resources that reference the policy.

    :param policy_arn: The ARN of the policy.
    :param actions: The actions to allow in the policy version.
    :param resource_arn: The ARN of the resource this policy version applies to.
    :param set_as_default: When True, this policy version is set as the default
                           version for the policy. Otherwise, the default
                           is not changed.
    :return: The newly created policy version.
    """
    policy_doc = {
        "Version": "2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
```

```
        policy = iam.Policy(policy_arn)
        policy_version = policy.create_version(
            PolicyDocument=json.dumps(policy_doc), SetAsDefault=set_as_default
        )
        logger.info(
            "Created policy version %s for policy %s.",
            policy_version.version_id,
            policy_version.arn,
        )
    except ClientError:
        logger.exception("Couldn't create a policy version for %s.", policy_arn)
        raise
    else:
        return policy_version
```

- Per i dettagli sull'API, consulta CreatePolicyVersion in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Gestione dei ruoli

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#####################################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####################################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####################################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
#####################################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
```

```
  echo "  -n role_name   The name of the IAM role."
  echo "  -p policy_json -- The assume role policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-role \
  --role-name "$role_name" \
  --assume-role-policy-document "$policy_document" \
  --output text \
  --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::createIamRole(
        const Aws::String &roleName,
        const Aws::String &policy,
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::CreateRoleRequest request;

    request.SetRoleName(roleName);
    request.SetAssumeRolePolicyDocument(policy);

    Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error creating role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const Aws::IAM::Model::Role iamRole = outcome.GetResult().GetRole();
        std::cout << "Created role " << iamRole.GetRoleName() << "\n";
        std::cout << "ID: " << iamRole.GetRoleId() << "\n";
```

```
        std::cout << "ARN: " << iamRole.GetArn() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare un ruolo IAM

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e collega una policy di attendibilità a tale ruolo.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Output:

```
{
    "Role": {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "CreateDate": "2013-06-07T20:43:32.821Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

La policy di attendibilità è definita come documento JSON nel file Test-Role-Trust-Policy.json. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per collegare una policy di autorizzazioni a un ruolo, usa il comando `put-role-policy`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

Esempio 2: Come creare un ruolo IAM con una durata massima della sessione specificata

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e imposta una durata massima della sessione di 7200 secondi (2 ore).

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --max-session-duration 7200
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:role/Test-Role",
        "CreateDate": "2023-05-24T23:50:25+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::12345678012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Per ulteriori informazioni, consulta Modificare la durata massima della sessione (AWS API) di un ruolo nella Guida per l'utente AWS IAM.

Esempio 3: Come creare un ruolo IAM con tag

Il comando seguente crea un ruolo IAM `Test-Role` con tag. Questo esempio utilizza il flag del parametro `--tags` con i seguenti tag in formato JSON: `'{"Key": "Department",`

`"Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`.
In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role",
        "CreateDate": "2023-05-25T23:29:41+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
```

```
}
```

Per ulteriori informazioni, consulta Applicazione di tag a ruoli IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta CreateRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(roleName string, trustedUserArn string)
 (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action: []string{"sts:AssumeRole"},
```

```
 }},
}
policyBytes, err := json.Marshal(trustPolicy)
if err != nil {
 log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
trustedUserArn, err)
 return nil, err
}
result, err := wrapper.IamClient.CreateRole(context.TODO(),
&iam.CreateRoleInput{
 AssumeRolePolicyDocument: aws.String(string(policyBytes)),
 RoleName:                 aws.String(roleName),
})
if err != nil {
 log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
} else {
 role = result.Role
}
return role, err
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import software.amazon.awssdk.services.iam.model.CreateRoleRequest;
import software.amazon.awssdk.services.iam.model.CreateRoleResponse;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
```

```
import java.io.FileReader;

/*
*   This example requires a trust policy document. For more information, see:
*   https://aws.amazon.com/blogs/security/how-to-use-trust-policies-with-iam-
roles/
*
*
*  In addition, set up your development environment, including your credentials.
*
*  For information, see this documentation topic:
*
*  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class CreateRole {
    public static void main(String[] args) throws Exception {
        final String usage = """
                Usage:
                    <rolename> <fileLocation>\s

                Where:
                    rolename - The name of the role to create.\s
                    fileLocation - The location of the JSON document that
 represents the trust policy.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String rolename = args[0];
        String fileLocation = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMRole(iam, rolename, fileLocation);
        System.out.println("Successfully created user: " + result);
        iam.close();
    }
```

```
   public static String createIAMRole(IamClient iam, String rolename, String
fileLocation) throws Exception {
       try {
           JSONObject jsonObject = (JSONObject)
readJsonSimpleDemo(fileLocation);
           CreateRoleRequest request = CreateRoleRequest.builder()
                   .roleName(rolename)
                   .assumeRolePolicyDocument(jsonObject.toJSONString())
                   .description("Created using the AWS SDK for Java")
                   .build();

           CreateRoleResponse response = iam.createRole(request);
           System.out.println("The ARN of the role is " +
response.role().arn());

       } catch (IamException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return "";
   }

   public static Object readJsonSimpleDemo(String filename) throws Exception {
       FileReader reader = new FileReader(filename);
       JSONParser jsonParser = new JSONParser();
       return jsonParser.parse(reader);
   }
}
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea il ruolo.

```
import { CreateRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const createRole = (roleName) => {
  const command = new CreateRoleCommand({
    AssumeRolePolicyDocument: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Effect: "Allow",
          Principal: {
            Service: "lambda.amazonaws.com",
          },
          Action: "sts:AssumeRole",
        },
      ],
    }),
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

    /**
     * @param string $roleName
     * @param string $rolePolicyDocument
     * @return array
     * @throws AwsException
     */
    public function createRole(string $roleName, string $rolePolicyDocument)
    {
        $result = $this->customWaiter(function () use ($roleName,
 $rolePolicyDocument) {
            return $this->iamClient->createRole([
                'AssumeRolePolicyDocument' => $rolePolicyDocument,
                'RoleName' => $roleName,
            ]);
        });
        return $result['Role'];
    }
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea un nuovo ruolo denominato **`MyNewRole`** e gli allega la politica trovata nel file**`NewRoleTrustPolicy.json`**. Si noti che è necessario utilizzare il parametro **`-Raw`** switch per elaborare correttamente il file di policy JSON. Il documento di policy visualizzato nell'output è codificato in URL. In questo esempio viene decodificato con il **`UrlDecode`** metodo.NET.

```
$results = New-IAMRole -AssumeRolePolicyDocument (Get-Content -raw
 NewRoleTrustPolicy.json) -RoleName MyNewRole
$results
```

Output:

```
Arn                       : arn:aws:iam::123456789012:role/MyNewRole
AssumeRolePolicyDocument : %7B%0D%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C
%0D%0A%20%20%22Statement%22
                           %3A%20%5B%0D%0A%20%20%20%20%7B%0D%0A
%20%20%20%20%20%20%22Sid%22%3A%20%22%22%2C
                           %0D%0A%20%20%20%20%20%20%22Effect%22%3A%20%22Allow
%22%2C%0D%0A%20%20%20%20%20%20
                           %22Principal%22%3A%20%7B%0D%0A
%20%20%20%20%20%20%20%20%22AWS%22%3A%20%22arn%3Aaws
                           %3Aiam%3A%3A123456789012%3ADavid%22%0D%0A
%20%20%20%20%20%20%7D%2C%0D%0A%20%20%20
                           %20%20%20%22Action%22%3A%20%22sts%3AAssumeRole%22%0D
%0A%20%20%20%20%7D%0D%0A%20
                           %20%5D%0D%0A%7D
CreateDate                : 4/15/2015 11:04:23 AM
Path                      : /
RoleId                    : V5PAJI2KPN4EAEXAMPLE1
RoleName                  : MyNewRole

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
```

```
[System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:David"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Per i dettagli sull'API, vedere CreateRolein AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_role(role_name, allowed_services):
    """
    Creates a role that lets a list of specified services assume the role.

    :param role_name: The name of the role.
    :param allowed_services: The services that can assume the role.
    :return: The newly created role.
    """
    trust_policy = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {"Service": service},
                "Action": "sts:AssumeRole",
```

```
            }
            for service in allowed_services
        ],
    }

    try:
        role = iam.create_role(
            RoleName=role_name, AssumeRolePolicyDocument=json.dumps(trust_policy)
        )
        logger.info("Created role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create role %s.", role_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta CreateRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates a role and attaches policies to it.
 #
 # @param role_name [String] The name of the role.
 # @param assume_role_policy_document [Hash] The trust relationship policy
document.
 # @param policy_arns [Array<String>] The ARNs of the policies to attach.
 # @return [String, nil] The ARN of the new role if successful, or nil if an
error occurred.
 def create_role(role_name, assume_role_policy_document, policy_arns)
   response = @iam_client.create_role(
     role_name: role_name,
```

```
      assume_role_policy_document: assume_role_policy_document.to_json
    )
    role_arn = response.role.arn

    policy_arns.each do |policy_arn|
      @iam_client.attach_role_policy(
        role_name: role_name,
        policy_arn: policy_arn
      )
    end

    role_arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating role: #{e.message}")
    nil
  end
```

- Per i dettagli sull'API, consulta la CreateRolesezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn create_role(
    client: &iamClient,
    role_name: &str,
    role_policy_document: &str,
) -> Result<Role, iamError> {
    let response: CreateRoleOutput = loop {
        if let Ok(response) = client
            .create_role()
            .role_name(role_name)
            .assume_role_policy_document(role_policy_document)
            .send()
            .await
```

```
        {
            break response;
        }
    };

    Ok(response.role.unwrap())
}
```

- Per i dettagli sulle API, consulta la CreateRoleguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public func createRole(name: String, policyDocument: String) async throws ->
String {
        let input = CreateRoleInput(
            assumeRolePolicyDocument: policyDocument,
            roleName: name
        )
        do {
            let output = try await client.createRole(input: input)
            guard let role = output.role else {
                throw ServiceHandlerError.noSuchRole
            }
            guard let id = role.roleId else {
                throw ServiceHandlerError.noSuchRole
```

```
            }
            return id
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateSAMLProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateSAMLProvider`.

CLI

AWS CLI

Come creare un provider SAML

Questo esempio crea un nuovo provider SAML in IAM denominato `MySAMLProvider`. È descritto dal documento di metadati SAML che si trova nel file `SAMLMetaData.xml`.

```
aws iam create-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --name MySAMLProvider
```

Output:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella Documentazione di riferimento dei comandi della AWS CLI .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { CreateSAMLProviderCommand, IAMClient } from "@aws-sdk/client-iam";
import { readFileSync } from "fs";
import * as path from "path";
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";

const client = new IAMClient({});

/**
 * This sample document was generated using Auth0.
 * For more information on generating this document,
 see https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_providers_create_saml.html#samlstep1.
 */
const sampleMetadataDocument = readFileSync(
  path.join(
    dirnameFromMetaUrl(import.meta.url),
    "../../../../resources/sample_files/sample_saml_metadata.xml",
  ),
);

/**
 *
 * @param {*} providerName
 * @returns
 */
export const createSAMLProvider = async (providerName) => {
  const command = new CreateSAMLProviderCommand({
    Name: providerName,
    SAMLMetadataDocument: sampleMetadataDocument.toString(),
  });

  const response = await client.send(command);
```

```
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta CreateSAMLProvider nella Documentazione di riferimento dell'API di AWS SDK for JavaScript .

PowerShell

Utensili per PowerShell

Esempio 1: questo esempio crea una nuova entità provider SAML in IAM. È denominata **`MySAMLProvider`** ed è descritta dal documento di metadati SAML presente nel file**`SAMLMetaData.xml`**, che è stato scaricato separatamente dal sito Web del provider di servizi SAML.

```
New-IAMSAMLProvider -Name MySAMLProvider -SAMLMetadataDocument (Get-Content -Raw
 SAMLMetaData.xml)
```

Output:

```
arn:aws:iam::123456789012:saml-provider/MySAMLProvider
```

- Per i dettagli sull'API, vedere CreateSAMLProvider in Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateServiceLinkedRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateServiceLinkedRole`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Come creare un ruolo collegato a un servizio

L'`create-service-linked-role`esempio seguente crea un ruolo collegato al servizio per il AWS servizio specificato e allega la descrizione specificata.

```
aws iam create-service-linked-role \
    --aws-service-name lex.amazonaws.com \
    --description "My service-linked role to support Lex"
```

Output:

```
{
    "Role": {
        "Path": "/aws-service-role/lex.amazonaws.com/",
        "RoleName": "AWSServiceRoleForLexBots",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots",
        "CreateDate": "2019-04-17T20:34:14+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "lex.amazonaws.com"
                        ]
                    }
                }
            ]
        }
    }
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, vedere CreateServiceLinkedRolein AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(serviceName string,
 description string) (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(context.TODO(),
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un ruolo collegato ai servizi.

```
import {
  CreateServiceLinkedRoleCommand,
  GetRoleCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} serviceName
 */
export const createServiceLinkedRole = async (serviceName) => {
  const command = new CreateServiceLinkedRoleCommand({
    // For a list of AWS services that support service-linked roles,
    // see https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-
services-that-work-with-iam.html.
    //
    // For a list of AWS service endpoints, see https://docs.aws.amazon.com/
general/latest/gr/aws-service-information.html.
    AWSServiceName: serviceName,
  });
  try {
    const response = await client.send(command);
    console.log(response);
    return response;
  } catch (caught) {
    if (
      caught instanceof Error &&
      caught.name === "InvalidInputException" &&
      caught.message.includes(
```

```
        "Service role name AWSServiceRoleForElasticBeanstalk has been taken in
 this account",
      )
    ) {
      console.warn(caught.message);
      return client.send(
        new GetRoleCommand({ RoleName: "AWSServiceRoleForElasticBeanstalk" }),
      );
    } else {
      throw caught;
    }
  }
};
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function createServiceLinkedRole($awsServiceName, $customSuffix = "",
 $description = "")
    {
        $createServiceLinkedRoleArguments = ['AWSServiceName' =>
 $awsServiceName];
        if ($customSuffix) {
            $createServiceLinkedRoleArguments['CustomSuffix'] = $customSuffix;
        }
        if ($description) {
            $createServiceLinkedRoleArguments['Description'] = $description;
```

```
        }
        return $this->iamClient-
>createServiceLinkedRole($createServiceLinkedRoleArguments);
    }
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea un ruolo collegato ai servizi per il servizio di scalabilità automatica.

```
New-IAMServiceLinkedRole -AWSServiceName autoscaling.amazonaws.com -CustomSuffix
 RoleNameEndsWithThis -Description "My service-linked role to support
 autoscaling"
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. CreateServiceLinkedRoleAWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_service_linked_role(service_name, description):
    """
    Creates a service-linked role.

    :param service_name: The name of the service that owns the role.
    :param description: A description to give the role.
    :return: The newly created role.
    """
```

```
    try:
        response = iam.meta.client.create_service_linked_role(
            AWSServiceName=service_name, Description=description
        )
        role = iam.Role(response["Role"]["RoleName"])
        logger.info("Created service-linked role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create service-linked role for %s.",
service_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta CreateServiceLinkedRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates a service-linked role
 #
 # @param service_name [String] The service name to create the role for.
 # @param description [String] The description of the service-linked role.
 # @param suffix [String] Suffix for customizing role name.
 # @return [String] The name of the created role
 def create_service_linked_role(service_name, description, suffix)
   response = @iam_client.create_service_linked_role(
     aws_service_name: service_name, description: description, custom_suffix:
suffix,)
   role_name = response.role.role_name
   @logger.info("Created service-linked role #{role_name}.")
```

```
    role_name
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't create service-linked role for #{service_name}.
 Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
```

- Per i dettagli sull'API, CreateServiceLinkedRoleconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn create_service_linked_role(
    client: &iamClient,
    aws_service_name: String,
    custom_suffix: Option<String>,
    description: Option<String>,
) -> Result<CreateServiceLinkedRoleOutput,
 SdkError<CreateServiceLinkedRoleError>> {
    let response = client
        .create_service_linked_role()
        .aws_service_name(aws_service_name)
        .set_custom_suffix(custom_suffix)
        .set_description(description)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento CreateServiceLinkedRoleall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public func createServiceLinkedRole(service: String, suffix: String? = nil,
description: String?)
                   async throws -> IAMClientTypes.Role {
       let input = CreateServiceLinkedRoleInput(
           awsServiceName: service,
           customSuffix: suffix,
           description: description
       )
       do {
           let output = try await client.createServiceLinkedRole(input: input)
           guard let role = output.role else {
               throw ServiceHandlerError.noSuchRole
           }
           return role
       } catch {
           throw error
       }
   }
```

- Per i dettagli sull'API, consulta la CreateServiceLinkedRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
```

```
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
```

```
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta CreateUser AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::CreateUserRequest create_request;
    create_request.SetUserName(userName);

    auto create_outcome = iam.CreateUser(create_request);
    if (!create_outcome.IsSuccess()) {
        std::cerr << "Error creating IAM user " << userName << ":" <<
                  create_outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully created IAM user " << userName << std::endl;
    }

    return create_outcome.IsSuccess();
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Esempio 1: Come creare un utente IAM

Il comando `create-user` seguente crea un utente IAM denominato Bob nell'account corrente.

```
aws iam create-user \
    --user-name Bob
```

Output:

```
{
    "User": {
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
```

```
}
```

Per ulteriori informazioni, consulta Creare un utente IAM nel tuo AWS account nella Guida per l'utente AWS IAM.

Esempio 2: Come creare un utente IAM in un percorso specificato

Il comando `create-user` seguente crea un utente IAM denominato Bob nel percorso specificato.

```
aws iam create-user \
    --user-name Bob \
    --path /division_abc/subdivision_xyz/
```

Output:

```
{
    "User": {
        "Path": "/division_abc/subdivision_xyz/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",
        "CreateDate": "2023-05-24T18:20:17+00:00"
    }
}
```

Per ulteriori informazioni, consulta Identificatori IAM nella Guida per l'utente di IAM AWS .

Esempio 3: Come creare un utente IAM con tag

Il comando `create-user` seguente crea un utente IAM denominato Bob con tag. Questo esempio utilizza il flag del parametro `--tags` con i seguenti tag in formato JSON: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-user \
    --user-name Bob \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

Output:

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-25T17:14:21+00:00",
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

Per ulteriori informazioni, consulta Applicazione di tag a utenti IAM nella Guida per l'utente di IAM AWS .

Esempio 3: Come creare un utente IAM con un limite delle autorizzazioni impostato

Il `create-user` comando seguente crea un utente IAM denominato Bob con il limite delle autorizzazioni di AmazonS3. FullAccess

```
aws iam create-user \
    --user-name Bob \
    --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

Output:

```
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::12345678012:user/Bob",
        "CreateDate": "2023-05-24T17:50:53+00:00",
```

```
        "PermissionsBoundary": {
        "PermissionsBoundaryType": "Policy",
        "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
        }
    }
}
```

Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta Command Reference. CreateUserAWS CLI

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(userName string) (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(context.TODO(),
 &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
```

```
 }
 return user, err
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.services.iam.model.CreateUserRequest;
import software.amazon.awssdk.services.iam.model.CreateUserResponse;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.waiters.IamWaiter;
import software.amazon.awssdk.services.iam.model.GetUserRequest;
import software.amazon.awssdk.services.iam.model.GetUserResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
```

```
                    <username>\s

                Where:
                    username - The name of the user to create.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String username = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        String result = createIAMUser(iam, username);
        System.out.println("Successfully created user: " + result);
        iam.close();
    }

    public static String createIAMUser(IamClient iam, String username) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();

            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            CreateUserResponse response = iam.createUser(request);

            // Wait until the user is created.
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
iamWaiter.waitUntilUserExists(userRequest);

waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user().userName();
```

```
        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare l'utente.

```
import { CreateUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} name
 */
export const createUser = (name) => {
  const command = new CreateUserCommand({ UserName: name });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, CreateUserconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  UserName: process.argv[2],
};

iam.getUser(params, function (err, data) {
  if (err && err.code === "NoSuchEntity") {
    iam.createUser(params, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Success", data);
      }
    });
  } else {
    console.log(
      "User " + process.argv[2] + " already exists",
      data.User.UserId
    );
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, CreateUserconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createIAMUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";
```

```
    /**
     * @param string $name
     * @return array
     * @throws AwsException
     */
    public function createUser(string $name): array
    {
        $result = $this->iamClient->createUser([
            'UserName' => $name,
        ]);

        return $result['User'];
    }
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea un utente IAM denominato**Bob**. Se Bob deve accedere alla AWS console, devi eseguire separatamente il comando **`New-IAMLoginProfile`** per creare un profilo di accesso con una password. Se Bob deve eseguire AWS PowerShell comandi CLI multipiattaforma o AWS effettuare chiamate API, è necessario eseguire separatamente **`New-IAMAccessKey`** il comando per creare le chiavi di accesso.

```
New-IAMUser -UserName Bob
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/22/2015 12:02:11 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : AIDAJWGEFDMEMEXAMPLE1
UserName         : Bob
```

- Per i dettagli sull'API, vedere CreateUserin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def create_user(user_name):
    """
    Creates a user. By default, a user has no permissions or access keys.

    :param user_name: The name of the user.
    :return: The newly created user.
    """
    try:
        user = iam.create_user(UserName=user_name)
        logger.info("Created user %s.", user.name)
    except ClientError:
        logger.exception("Couldn't create user %s.", user_name)
        raise
    else:
        return user
```

- Per i dettagli sull'API, consulta CreateUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  # Creates a user and their login profile
  #
  # @param user_name [String] The name of the user
  # @param initial_password [String] The initial password for the user
  # @return [String, nil] The ID of the user if created, or nil if an error
 occurred
  def create_user(user_name, initial_password)
    response = @iam_client.create_user(user_name: user_name)
    @iam_client.wait_until(:user_exists, user_name: user_name)
    @iam_client.create_login_profile(
      user_name: user_name,
      password: initial_password,
      password_reset_required: true
    )
    @logger.info("User '#{user_name}' created successfully.")
    response.user.user_id
  rescue Aws::IAM::Errors::EntityAlreadyExists
    @logger.error("Error creating user '#{user_name}': user already exists.")
    nil
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating user '#{user_name}': #{e.message}")
    nil
  end
```

- Per i dettagli sull'API, CreateUserconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn create_user(client: &iamClient, user_name: &str) -> Result<User,
 iamError> {
    let response = client.create_user().user_name(user_name).send().await?;
```

```
    Ok(response.user.unwrap())
}
```

- Per i dettagli sulle API, consulta il riferimento CreateUserall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func createUser(name: String) async throws -> String {
        let input = CreateUserInput(
            userName: name
        )
        do {
            let output = try await client.createUser(input: input)
            guard let user = output.user else {
                throw ServiceHandlerError.noSuchUser
            }
            guard let id = user.userId else {
                throw ServiceHandlerError.noSuchUser
            }
            return id
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la CreateUserguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateVirtualMfaDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateVirtualMfaDevice`.

CLI

AWS CLI

Per creare un dispositivo MFA virtuale

Questo esempio crea un nuovo dispositivo MFA virtuale chiamato. `BobsMFADevice` Crea un file che contiene le informazioni di bootstrap richiamate `QRCode.png` e le inserisce nella `C:/` directory. Il metodo bootstrap utilizzato in questo esempio è. `QRCodePNG`

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

Output:

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta CreateVirtualMfaDevice AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il **`Base32StringSeed`** valore necessario al programma software MFA virtuale per creare un account (in alternativa al codice QR). Dopo aver configurato il programma con il valore, ottieni due codici di autenticazione sequenziali dal programma. Infine, utilizza l'ultimo comando per collegare il dispositivo MFA virtuale all'utente IAM **`Bob`** e sincronizzare l'account con i due codici di autenticazione.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$SR = New-Object System.IO.StreamReader($Device.Base32StringSeed)
$base32stringseed = $SR.ReadToEnd()
$base32stringseed
CZWZMCQNW4DEXAMPLE3VOUGXJFZYSUW7EXAMPLECR4NJFD65GX2SLUDW2EXAMPLE
```

Output:

```
-- Pause here to enter base-32 string seed code into virtual MFA program to
 register account. --

Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

Esempio 2: questo esempio crea un nuovo dispositivo MFA virtuale. Le righe 2 e 3 estraggono il **`QRCodePNG`** valore e lo scrivono in un file. Questa immagine può essere scansionata dal programma software MFA virtuale per creare un account (in alternativa all'immissione manuale del valore StringSeed Base32). Dopo aver creato l'account nel tuo programma MFA virtuale, ottieni due codici di autenticazione sequenziali e inseriscili negli ultimi comandi per collegare il dispositivo MFA virtuale all'utente **`Bob`** IAM e sincronizzare l'account.

```
$Device = New-IAMVirtualMFADevice -VirtualMFADeviceName BobsMFADevice
$BR = New-Object System.IO.BinaryReader($Device.QRCodePNG)
$BR.ReadBytes($BR.BaseStream.Length) | Set-Content -Encoding Byte -Path
 QRCode.png
```

Output:

```
 -- Pause here to scan PNG with virtual MFA program to register account. --
```

```
Enable-IAMMFADevice -SerialNumber $Device.SerialNumber -UserName Bob -
AuthenticationCode1 123456 -AuthenticationCode2 789012
```

- Per i dettagli sull'API, vedere CreateVirtualMfaDevicein Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeactivateMfaDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeactivateMfaDevice`.

CLI

AWS CLI

Per disattivare un dispositivo MFA

Questo comando disattiva il dispositivo MFA virtuale con l'ARN `arn:aws:iam::210987654321:mfa/BobsMFADevice` associato all'utente. `Bob`

```
aws iam deactivate-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta DeactivateMfaDevice in Command Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando disabilita il dispositivo hardware MFA associato all'**Bob**utente con il numero di serie. **123456789012**

```
Disable-IAMMFADevice -UserName "Bob" -SerialNumber "123456789012"
```

Esempio 2: questo comando disabilita il dispositivo MFA virtuale associato all'**David**utente che dispone dell'ARN. **arn:aws:iam::210987654321:mfa/David** Tieni presente che il dispositivo MFA virtuale non viene eliminato dall'account. Il dispositivo virtuale è ancora presente e appare nell'output del **Get-IAMVirtualMFADevice** comando. Prima di poter creare un nuovo dispositivo MFA virtuale per lo stesso utente, è necessario eliminare quello precedente utilizzando il **Remove-IAMVirtualMFADevice** comando.

```
Disable-IAMMFADevice -UserName "David" -SerialNumber
 "arn:aws:iam::210987654321:mfa/David"
```

- Per i dettagli sull'API, vedere DeactivateMfaDevice in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteAccessKey** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle chiavi di accesso

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an WS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
```

```
  echo "  -k access_key   The access key to delete."
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:k:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    k) access_key="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

if [[ -z "$access_key" ]]; then
  errecho "ERROR: You must provide an access key with the -k parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Username:   $user_name"
iecho "    Access key:   $access_key"
iecho ""

response=$(aws iam delete-access-key \
  --user-name "$user_name" \
  --access-key-id "$access_key")

local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteAccessKey(const Aws::String &userName,
                                  const Aws::String &accessKeyID,
                                  const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteAccessKeyRequest request;
    request.SetUserName(userName);
    request.SetAccessKeyId(accessKeyID);

    auto outcome = iam.DeleteAccessKey(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting access key " << accessKeyID << " from user "
                  << userName << ": " << outcome.GetError().GetMessage() <<
                  std::endl;
```

```
    }
    else {
        std::cout << "Successfully deleted access key " << accessKeyID
                  << " for IAM user " << userName << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come eliminare una chiave di accesso per un utente IAM

Il comando `delete-access-key` seguente elimina la chiave di accesso specificata (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam delete-access-key \
    --access-key-id AKIDPMS9RO4H3FEXAMPLE \
    --user-name Bob
```

Questo comando non produce alcun output.

Per elencare le chiavi di accesso definite per un utente IAM, usa il comando `list-access-keys`.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteAccessKey deletes an access key from a user.
func (wrapper UserWrapper) DeleteAccessKey(userName string, keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(context.TODO(),
 &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.DeleteAccessKeyRequest;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteAccessKey {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <username> <accessKey>\s

                Where:
                    username - The name of the user.\s
                    accessKey - The access key ID for the secret access key you
 want to delete.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String username = args[0];
        String accessKey = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();
        deleteKey(iam, username, accessKey);
        iam.close();
    }

    public static void deleteKey(IamClient iam, String username, String
 accessKey) {
        try {
            DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                    .accessKeyId(accessKey)
                    .userName(username)
                    .build();

            iam.deleteAccessKey(request);
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina la chiave di accesso.

```
import { DeleteAccessKeyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 * @param {string} accessKeyId
 */
export const deleteAccessKey = (userName, accessKeyId) => {
  const command = new DeleteAccessKeyCommand({
    AccessKeyId: accessKeyId,
    UserName: userName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  AccessKeyId: "ACCESS_KEY_ID",
```

```
  UserName: "USER_NAME",
};

iam.deleteAccessKey(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteKey(
    userNameVal: String,
    accessKey: String
) {
    val request =
        DeleteAccessKeyRequest {
            accessKeyId = accessKey
            userName = userNameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteAccessKey(request)
        println("Successfully deleted access key $accessKey from $userNameVal")
    }
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio elimina la coppia di chiavi di AWS accesso con l'ID della chiave **`AKIAIOSFODNN7EXAMPLE`** dall'utente denominato**`Bob`**.

```
Remove-IAMAccessKey -AccessKeyId AKIAIOSFODNN7EXAMPLE -UserName Bob -Force
```

- Per i dettagli sull'API, vedere DeleteAccessKey in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity => e
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
```

```
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded => e
    @logger.error("Error creating access key: limit exceeded. Cannot create
more.")
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: "Inactive"
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
```

```
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_access_key(
    client: &iamClient,
    user: &User,
    key: &AccessKey,
) -> Result<(), iamError> {
    loop {
        match client
            .delete_access_key()
            .user_name(user.user_name())
            .access_key_id(key.access_key_id())
            .send()
            .await
        {
            Ok(_) => {
                break;
            }
            Err(e) => {
                println!("Can't delete the access key: {:?}", e);
```

```
                sleep(Duration::from_secs(2)).await;
            }
        }
    }
    Ok(())
}
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func deleteAccessKey(user: IAMClientTypes.User? = nil,
                                key: IAMClientTypes.AccessKey) async throws {
        let userName: String?

        if user != nil {
            userName = user!.userName
        } else {
            userName = nil
        }

        let input = DeleteAccessKeyInput(
            accessKeyId: key.accessKeyId,
            userName: userName
        )
```

```
        do {
            _ = try await iamClient.deleteAccessKey(input: input)
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta DeleteAccessKey in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteAccountAlias`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccountAlias`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteAccountAlias(const Aws::String &accountAlias,
                                     const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteAccountAliasRequest request;
    request.SetAccountAlias(accountAlias);
```

```
    const auto outcome = iam.DeleteAccountAlias(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting account alias " << accountAlias << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully deleted account alias " << accountAlias <<
                  std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come eliminare l'alias di un account

Il comando `delete-account-alias` seguente rimuove l'alias `mycompany` per l'account corrente.

```
aws iam delete-account-alias \
    --account-alias mycompany
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta l'ID AWS dell'account e il suo alias nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DeleteAccountAliasRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteAccountAlias {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <alias>\s

                Where:
                    alias - The account alias to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alias = args[0];
        Region region = Region.AWS_GLOBAL;
```

```
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMAccountAlias(iam, alias);
        iam.close();
    }

    public static void deleteIAMAccountAlias(IamClient iam, String alias) {
        try {
            DeleteAccountAliasRequest request =
 DeleteAccountAliasRequest.builder()
                    .accountAlias(alias)
                    .build();

            iam.deleteAccountAlias(request);
            System.out.println("Successfully deleted account alias " + alias);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina l'alias dell'account.

```
import { DeleteAccountAliasCommand, IAMClient } from "@aws-sdk/client-iam";
```

```
const client = new IAMClient({});

/**
 *
 * @param {string} alias
 */
export const deleteAccountAlias = (alias) => {
  const command = new DeleteAccountAliasCommand({ AccountAlias: alias });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.deleteAccountAlias({ AccountAlias: process.argv[2] }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteIAMAccountAlias(alias: String) {
    val request =
        DeleteAccountAliasRequest {
            accountAlias = alias
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteAccountAlias(request)
        println("Successfully deleted account alias $alias")
    }
}
```

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio rimuove l'alias dell'account dal tuo Account AWS. La pagina di accesso utente con l'alias all'indirizzo https://mycompanyaws.signin.aws.amazon.com/console non funziona più. Devi invece utilizzare l'URL originale con il tuo numero Account AWS ID su https://.signin.aws.amazon.com/console. <accountidnumber>

```
Remove-IAMAccountAlias -AccountAlias mycompanyaws
```

- Per i dettagli sull'API, consulta Alias in Cmdlet Reference. DeleteAccount AWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def delete_alias(alias):
    """
    Removes the alias from the current account.

    :param alias: The alias to remove.
    """
    try:
        iam.meta.client.delete_account_alias(AccountAlias=alias)
        logger.info("Removed alias '%s' from your account.", alias)
    except ClientError:
        logger.exception("Couldn't remove alias '%s' from your account.", alias)
        raise
```

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info("Account aliases are:")
      response.account_aliases.each { |account_alias| @logger.info("
 #{account_alias}") }
    else
      @logger.info("No account aliases found.")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
```

```
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta DeleteAccountAlias in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteAccountPasswordPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAccountPasswordPolicy`.

CLI

AWS CLI

Per eliminare la politica corrente in materia di password dell'account

Il `delete-account-password-policy` comando seguente rimuove la politica relativa alle password per l'account corrente.

```
aws iam delete-account-password-policy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, vedere DeleteAccountPasswordPolicyin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio elimina la politica relativa alle password per Account AWS e ripristina tutti i valori ai valori predefiniti originali. Se attualmente non esiste una politica in materia di password, viene visualizzato il seguente messaggio di errore: Impossibile trovare la politica dell'account con il nome PasswordPolicy .

```
Remove-IAMAccountPasswordPolicy
```

- Per i dettagli sull'API, vedere DeleteAccountPasswordPolicyin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupAsync(string groupName)
    {
        var response = await _IAMService.DeleteGroupAsync(new DeleteGroupRequest
{ GroupName = groupName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteGroupsezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Come eliminare un gruppo IAM

Il comando `delete-group` seguente elimina un gruppo IAM denominato `MyTestGroup`.

```
aws iam delete-group \
    --group-name MyTestGroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Eliminazione di un gruppo di utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { DeleteGroupCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} groupName
 */
export const deleteGroup = async (groupName) => {
  const command = new DeleteGroupCommand({
    GroupName: groupName,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per i dettagli sull'API, consulta la DeleteGroupsezione AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il gruppo IAM denominato**`MyTestGroup`**. Il primo comando rimuove tutti gli utenti IAM che sono membri del gruppo e il secondo comando elimina il gruppo IAM. Entrambi i comandi funzionano senza alcuna richiesta di conferma.

```
(Get-IAMGroup -GroupName MyTestGroup).Users | Remove-IAMUserFromGroup -GroupName
 MyTestGroup -Force
Remove-IAMGroup -GroupName MyTestGroup -Force
```

- Per i dettagli sull'API, vedere DeleteGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteGroupPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteGroupPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM policy associated with an IAM group.
```

```
    /// </summary>
    /// <param name="groupName">The name of the IAM group associated with the
    /// policy.</param>
    /// <param name="policyName">The name of the policy to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupPolicyAsync(string groupName, string
 policyName)
    {
        var request = new DeleteGroupPolicyRequest()
        {
            GroupName = groupName,
            PolicyName = policyName,
        };

        var response = await _IAMService.DeleteGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteGroupPolicy in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come eliminare una policy da un gruppo IAM

Il comando `delete-group-policy` seguente elimina la policy denominata `ExamplePolicy` dal gruppo denominato `Admins`.

```
aws iam delete-group-policy \
    --group-name Admins \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per visualizzare le policy collegate a un gruppo, usa il comando `list-group-policies`.

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DeleteGroupPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio rimuove la policy in linea denominata **`TesterPolicy`** dal gruppo **`Testers`** IAM. Gli utenti di quel gruppo perdono immediatamente le autorizzazioni definite in quella politica.

```
Remove-IAMGroupPolicy -GroupName Testers -PolicyName TestPolicy
```

- Per i dettagli sull'API, vedere DeleteGroupPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteInstanceProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteInstanceProfile`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione e gestione di un servizio resiliente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Detaches a role from an instance profile, detaches policies from the
role,
    /// and deletes all the resources.
```

```
    /// </summary>
    /// <param name="profileName">The name of the profile to delete.</param>
    /// <param name="roleName">The name of the role to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteInstanceProfile(string profileName, string roleName)
    {
        try
        {
            await _amazonIam.RemoveRoleFromInstanceProfileAsync(
                new RemoveRoleFromInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
            await _amazonIam.DeleteInstanceProfileAsync(
                new DeleteInstanceProfileRequest() { InstanceProfileName =
profileName });
            var attachedPolicies = await
_amazonIam.ListAttachedRolePoliciesAsync(
                new ListAttachedRolePoliciesRequest() { RoleName = roleName });
            foreach (var policy in attachedPolicies.AttachedPolicies)
            {
                await _amazonIam.DetachRolePolicyAsync(
                    new DetachRolePolicyRequest()
                    {
                        RoleName = roleName,
                        PolicyArn = policy.PolicyArn
                    });
                // Delete the custom policies only.
                if (!policy.PolicyArn.StartsWith("arn:aws:iam::aws"))
                {
                    await _amazonIam.DeletePolicyAsync(
                        new Amazon.IdentityManagement.Model.DeletePolicyRequest()
                        {
                            PolicyArn = policy.PolicyArn
                        });
                }
            }

            await _amazonIam.DeleteRoleAsync(
                new DeleteRoleRequest() { RoleName = roleName });
        }
        catch (NoSuchEntityException)
        {
```

```
            Console.WriteLine($"Instance profile {profileName} does not exist.");
        }
    }
```

- Per i dettagli sull'API, consulta DeleteInstanceProfile in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come eliminare un profilo dell'istanza

Il comando `delete-instance-profile` seguente elimina un profilo dell'istanza denominato `ExampleInstanceProfile`.

```
aws iam delete-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteInstanceProfile in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  const client = new IAMClient({});
  await client.send(
    new DeleteInstanceProfileCommand({
      InstanceProfileName: NAMES.instanceProfileName,
    }),
```

```
      );
```

- Per i dettagli sull'API, consulta DeleteInstanceProfile in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il profilo di istanza EC2 denominato. **`MyAppInstanceProfile`** Il primo comando rimuove tutti i ruoli dal profilo dell'istanza, quindi il secondo comando elimina il profilo dell'istanza.

```
(Get-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile).Roles |
 Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyAppInstanceProfile
Remove-IAMInstanceProfile -InstanceProfileName MyAppInstanceProfile
```

- Per i dettagli sull'API, vedere DeleteInstanceProfile in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo esempio rimuove il ruolo dal profilo dell'istanza, scollega tutte le policy collegate al ruolo ed elimina tutte le risorse.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
```

```
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
that is
                          created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
        self.key_pair_name = f"{resource_prefix}-key-pair"


    def delete_instance_profile(self, profile_name, role_name):
        """
        Detaches a role from an instance profile, detaches policies from the
role,
        and deletes all the resources.
```

```
        :param profile_name: The name of the profile to delete.
        :param role_name: The name of the role to delete.
        """
        try:
            self.iam_client.remove_role_from_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )

self.iam_client.delete_instance_profile(InstanceProfileName=profile_name)
            log.info("Deleted instance profile %s.", profile_name)
            attached_policies = self.iam_client.list_attached_role_policies(
                RoleName=role_name
            )
            for pol in attached_policies["AttachedPolicies"]:
                self.iam_client.detach_role_policy(
                    RoleName=role_name, PolicyArn=pol["PolicyArn"]
                )
                if not pol["PolicyArn"].startswith("arn:aws:iam::aws"):
                    self.iam_client.delete_policy(PolicyArn=pol["PolicyArn"])
                log.info("Detached and deleted policy %s.", pol["PolicyName"])
            self.iam_client.delete_role(RoleName=role_name)
            log.info("Deleted role %s.", role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "NoSuchEntity":
                log.info(
                    "Instance profile %s doesn't exist, nothing to do.",
profile_name
                )
            else:
                raise AutoScalerError(
                    f"Couldn't delete instance profile {profile_name} or detach "
                    f"policies and delete role {role_name}: {err}"
                )
```

- Per i dettagli sull'API, consulta DeleteInstanceProfile in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteLoginProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteLoginProfile`.

CLI

AWS CLI

Per eliminare una password per un utente IAM

Il `delete-login-profile` comando seguente elimina la password per l'utente IAM denominato`Bob`.

```
aws iam delete-login-profile \
    --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Managing password for IAM users nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta DeleteLoginProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il profilo di accesso dall'utente IAM denominato**`Bob`**. Ciò impedisce all'utente di accedere alla console. AWS Non impedisce all'utente di eseguire chiamate AWS CLI o API utilizzando chiavi di AWS accesso che potrebbero essere ancora collegate all'account utente. PowerShell

```
Remove-IAMLoginProfile -UserName Bob
```

- Per i dettagli sull'API, vedere DeleteLoginProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteOpenIdConnectProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteOpenIdConnectProvider`.

CLI

AWS CLI

Per eliminare un provider di identità IAM OpenID Connect

Questo esempio elimina il provider IAM OIDC che si connette al provider. `example.oidcprovider.com`

```
aws iam delete-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta DeleteOpenIdConnectProvider in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il provider IAM OIDC che si connette al provider. **`example.oidcprovider.com`** Assicurati di aggiornare o eliminare tutti i ruoli che fanno riferimento a questo provider nell'**`Principal`**elemento della politica di fiducia del ruolo.

```
Remove-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com
```

- Per i dettagli sull'API, vedere DeleteOpenIdConnectProvider in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeletePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeletePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura
- Gestione delle policy

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM policy.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeletePolicy AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deletePolicy(const Aws::String &policyArn,
                               const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::DeletePolicyRequest request;
    request.SetPolicyArn(policyArn);

    auto outcome = iam.DeletePolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting policy with arn " << policyArn << ": "
                  << outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully deleted policy with arn " << policyArn
```

```
                    << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Come eliminare una policy IAM

Questo esempio elimina la policyu il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeletePolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
```

```
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(policyArn string) error {
 _, err := wrapper.IamClient.DeletePolicy(context.TODO(), &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DeletePolicyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeletePolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <policyARN>\s

                Where:
                    policyARN - A policy ARN value to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String policyARN = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMPolicy(iam, policyARN);
        iam.close();
    }

    public static void deleteIAMPolicy(IamClient iam, String policyARN) {
        try {
            DeletePolicyRequest request = DeletePolicyRequest.builder()
                    .policyArn(policyARN)
                    .build();

            iam.deletePolicy(request);
            System.out.println("Successfully deleted the policy");

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
        System.out.println("Done");
    }
}
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Eliminare il criterio.

```
import { DeletePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 */
export const deletePolicy = (policyArn) => {
  const command = new DeletePolicyCommand({ PolicyArn: policyArn });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeletePolicysezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteIAMPolicy(policyARNVal: String?) {
    val request =
        DeletePolicyRequest {
            policyArn = policyARNVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deletePolicy(request)
        println("Successfully deleted $policyARNVal")
    }
}
```

- Per i dettagli sull'API, DeletePolicyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina la politica il cui **`arn:aws:iam::123456789012:policy/MySamplePolicy`** ARN è. Prima di poter eliminare la policy, è necessario eliminare tutte le versioni tranne quella predefinita eseguendo. **`Remove-IAMPolicyVersion`** È inoltre necessario scollegare la policy da qualsiasi utente, gruppo o ruolo IAM.

```
Remove-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Esempio 2: Questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy, scollegandola da tutte le entità IAM collegate e infine eliminando la

policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni delle politiche che non sono contrassegnate come versione predefinita in una raccolta, quindi elimina ogni politica nella raccolta. La terza riga recupera tutti gli utenti, i gruppi e i ruoli IAM a cui è associata la policy. Le righe da quattro a sei separano la policy da ogni entità collegata. L'ultima riga utilizza questo comando per rimuovere la politica gestita e la versione predefinita rimanente. L'esempio include il parametro **`-Force`** switch su qualsiasi riga che lo richieda per sopprimere le richieste di conferma.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
$attached = Get-IAMEntitiesForPolicy -PolicyArn $pol.Arn
$attached.PolicyGroups | Unregister-IAMGroupPolicy -PolicyArn $pol.arn
$attached.PolicyRoles | Unregister-IAMRolePolicy -PolicyArn $pol.arn
$attached.PolicyUsers | Unregister-IAMUserPolicy -PolicyArn $pol.arn
Remove-IAMPolicy $pol.Arn -Force
```

- Per i dettagli sull'API, vedere DeletePolicyin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
```

```
        raise
```

- Per i dettagli sull'API, consulta DeletePolicy AWSSDK for Python (Boto3) API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_policy(client: &iamClient, policy: Policy) -> Result<(),
 iamError> {
    client
        .delete_policy()
        .policy_arn(policy.arn.unwrap())
        .send()
        .await?;
    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeletePolicyguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func deletePolicy(policy: IAMClientTypes.Policy) async throws {
    let input = DeletePolicyInput(
        policyArn: policy.arn
    )
    do {
        _ = try await iamClient.deletePolicy(input: input)
    } catch {
        throw error
    }
}
```

- Per i dettagli sull'API, consulta la DeletePolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeletePolicyVersion`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeletePolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Per eliminare una versione di una policy gestita

Questo esempio elimina la versione identificata come v2 dalla policy il cui `arn:aws:iam::123456789012:policy/MySamplePolicy` ARN è.

```
aws iam delete-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeletePolicyVersion in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina la versione identificata come **v2** dalla policy il cui **arn:aws:iam::123456789012:policy/MySamplePolicy** ARN è.

```
Remove-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MySamplePolicy -VersionID v2
```

Esempio 2: Questo esempio elimina una policy eliminando prima tutte le versioni non predefinite della policy e quindi eliminando la policy stessa. La prima riga recupera l'oggetto della policy. La seconda riga recupera tutte le versioni delle politiche che non sono contrassegnate come predefinite in una raccolta e quindi utilizza questo comando per eliminare ogni politica nella raccolta. L'ultima riga rimuove la policy stessa e la versione predefinita rimanente. Tieni presente che per eliminare correttamente una policy gestita, devi anche scollegare la policy da qualsiasi utente, gruppo o ruolo utilizzando i **Unregister-IAMRolePolicy** comandi **Unregister-IAMUserPolicyUnregister-IAMGroupPolicy**, and. Vedere l'esempio per il **Remove-IAMPolicy** cmdlet.

```
$pol = Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
Get-IAMPolicyVersions -PolicyArn $pol.Arn | where {-not $_.IsDefaultVersion} |
 Remove-IAMPolicyVersion -PolicyArn $pol.Arn -force
Remove-IAMPolicy -PolicyArn $pol.Arn -force
```

- Per i dettagli sull'API, vedere DeletePolicyVersion in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un utente e assunzione di un ruolo
- Gestione dei ruoli

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
{ RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
```

```
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an WS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""
```

```
  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteRole AWS CLICommand Reference.

CLI

AWS CLI

Come eliminare un ruolo IAM

Il comando `delete-role` seguente rimuove il ruolo denominato `Test-Role`.

```
aws iam delete-role \
    --role-name Test-Role
```

Questo comando non produce alcun output.

Per poter eliminare un ruolo, devi prima rimuovere il ruolo da qualunque profilo dell'istanza (`remove-role-from-instance-profile`), scollegare eventuali policy gestite (`detach-role-policy`) ed eliminare tutte le policy inline collegate al ruolo (`delete-role-policy`).

Per ulteriori informazioni, consulta Ruoli IAM e Utilizzo dei profili dell'istanza nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta DeleteRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DeleteRole deletes a role. All attached policies must be detached before a
// role can be deleted.
func (wrapper RoleWrapper) DeleteRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteRole(context.TODO(), &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina il ruolo.

```
import { DeleteRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const deleteRole = (roleName) => {
  const command = new DeleteRoleCommand({ RoleName: roleName });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il ruolo denominato **`MyNewRole`** dall'account IAM corrente. Prima di poter eliminare il ruolo, devi prima utilizzare il **`Unregister-IAMRolePolicy`** comando per scollegare eventuali politiche gestite. Le politiche in linea vengono eliminate con il ruolo.

```
Remove-IAMRole -RoleName MyNewRole
```

Esempio 2: questo esempio rimuove tutte le politiche gestite dal ruolo denominato **MyNewRole** e quindi elimina il ruolo. La prima riga recupera tutte le policy gestite associate al ruolo come raccolta e quindi scollega ogni policy della raccolta dal ruolo. La seconda riga elimina il ruolo stesso. Le politiche in linea vengono eliminate insieme al ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName MyNewRole | Unregister-IAMRolePolicy -
RoleName MyNewRole
Remove-IAMRole -RoleName MyNewRole
```

- Per i dettagli sull'API, vedere DeleteRolein AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def delete_role(role_name):
    """
    Deletes a role.

    :param role_name: The name of the role to delete.
    """
    try:
        iam.Role(role_name).delete()
        logger.info("Deleted role %s.", role_name)
    except ClientError:
        logger.exception("Couldn't delete role %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  # Deletes a role and its attached policies.
  #
  # @param role_name [String] The name of the role to delete.
  def delete_role(role_name)
    begin
      # Detach and delete attached policies
      @iam_client.list_attached_role_policies(role_name: role_name).each do |
response|
        response.attached_policies.each do |policy|
          @iam_client.detach_role_policy({
                                           role_name: role_name,
                                           policy_arn: policy.policy_arn
                                         })
          # Check if the policy is a customer managed policy (not AWS managed)
          unless policy.policy_arn.include?("aws:policy/")
            @iam_client.delete_policy({ policy_arn: policy.policy_arn })
            @logger.info("Deleted customer managed policy
 #{policy.policy_name}.")
          end
        end
      end

      # Delete the role
      @iam_client.delete_role({ role_name: role_name })
      @logger.info("Deleted role #{role_name}.")
    rescue Aws::IAM::Errors::ServiceError => e
      @logger.error("Couldn't detach policies and delete role #{role_name}.
 Here's why:")
      @logger.error("\t#{e.code}: #{e.message}")
      raise
    end
  end
```

- Per i dettagli sull'API, consulta la DeleteRolesezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_role(client: &iamClient, role: &Role) -> Result<(), iamError>
 {
    let role = role.clone();
    while client
        .delete_role()
        .role_name(role.role_name())
        .send()
        .await
        .is_err()
    {
        sleep(Duration::from_secs(2)).await;
    }
    Ok(())
}
```

- Per i dettagli sulle API, consulta la DeleteRoleguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func deleteRole(role: IAMClientTypes.Role) async throws {
    let input = DeleteRoleInput(
        roleName: role.roleName
    )
    do {
        _ = try await iamClient.deleteRole(input: input)
    } catch {
        throw error
    }
}
```

- Per i dettagli sull'API, consulta la DeleteRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteRolePermissionsBoundary`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRolePermissionsBoundary`.

CLI

AWS CLI

Per eliminare un limite di autorizzazioni da un ruolo IAM

L'`delete-role-permissions-boundary`esempio seguente elimina il limite delle autorizzazioni per il ruolo IAM specificato. Per applicare un limite di autorizzazioni a un ruolo, usa il comando. `put-role-permissions-boundary`

```
aws iam delete-role-permissions-boundary \
    --role-name lambda-application-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command DeleteRolePermissionsBoundaryReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio mostra come rimuovere il limite di autorizzazione associato a un ruolo IAM.

```
Remove-IAMRolePermissionsBoundary -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere DeleteRolePermissionsBoundaryin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRolePolicy`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteRolePolicy in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come rimuovere una policy da un ruolo IAM

Il comando `delete-role-policy` seguente rimuove la policy denominata `ExamplePolicy` dal ruolo denominato `Test-Role`.

```
aws iam delete-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteRolePolicy in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { DeleteRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 * @param {string} policyName
 */
export const deleteRolePolicy = (roleName, policyName) => {
  const command = new DeleteRolePolicyCommand({
    RoleName: roleName,
    PolicyName: policyName,
  });
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteRolePolicy in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina la policy in linea **`S3AccessPolicy`** incorporata nel ruolo IAM. **`S3BackupRole`**

```
Remove-IAMRolePolicy -PolicyName S3AccessPolicy -RoleName S3BackupRole
```

- Per i dettagli sull'API, vedere DeleteRolePolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteSAMLProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteSAMLProvider`.

CLI

AWS CLI

Come eliminare un provider SAML

Questo esempio elimina il provider SAML 2.0 IAM il cui ARN è `arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`.

```
aws iam delete-saml-provider \
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella Documentazione di riferimento dei comandi della AWS CLI .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { DeleteSAMLProviderCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} providerArn
 * @returns
 */
export const deleteSAMLProvider = async (providerArn) => {
  const command = new DeleteSAMLProviderCommand({
    SAMLProviderArn: providerArn,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta DeleteSAMLProvider nella Documentazione di riferimento dell'API di AWS SDK for JavaScript .

PowerShell

Utensili per PowerShell

Esempio 1: questo esempio elimina il provider IAM SAML 2.0 il cui ARN è. **`arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`**

```
Remove-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFSProvider
```

- Per i dettagli sull'API, vedere DeleteSAMLProvider in Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteServerCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::deleteServerCertificate(const Aws::String &certificateName,
                                          const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::DeleteServerCertificateRequest request;
    request.SetServerCertificateName(certificateName);

    const auto outcome = iam.DeleteServerCertificate(request);
    bool result = true;
```

```
    if (!outcome.IsSuccess()) {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error deleting server certificate " << certificateName
 <<
                      ": " << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << certificateName
                      << "' not found." << std::endl;
        }
    }
    else {
        std::cout << "Successfully deleted server certificate " <<
 certificateName
                  << std::endl;
    }

    return result;
}
```

- Per i dettagli sull'API, consulta DeleteServerCertificate in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per eliminare un certificato server dal tuo AWS account

Il `delete-server-certificate` comando seguente rimuove il certificato del server specificato dal tuo AWS account.

```
aws iam delete-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Questo comando non produce alcun output.

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DeleteServerCertificate in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un certificato del server.

```
import { DeleteServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} certName
 */
export const deleteServerCertificate = (certName) => {
  const command = new DeleteServerCertificateCommand({
    ServerCertificateName: certName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteServerCertificate in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.deleteServerCertificate(
  { ServerCertificateName: "CERTIFICATE_NAME" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta DeleteServerCertificate in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio elimina il certificato del server denominato**`MyServerCert`**.

```
Remove-IAMServerCertificate -ServerCertificateName MyServerCert
```

- Per i dettagli sull'API, vedere DeleteServerCertificate in AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati del server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key,
                                          })
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates
```

```
    if response.server_certificate_metadata_list.empty?
      @logger.info("No server certificates found.")
      return
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
 #{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
 '#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta DeleteServerCertificate in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteServiceLinkedRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteServiceLinkedRole`.

CLI

AWS CLI

Come eliminare un ruolo collegato a un servizio

L'esempio `delete-service-linked-role` seguente elimina il ruolo collegato al servizio specificato che non è più necessario. L'eliminazione avviene in modo asincrono. Puoi anche controllare lo stato dell'eliminazione e confermare quando è stata completata utilizzando il comando `get-service-linked-role-deletion-status`.

```
aws iam delete-service-linked-role \
    --role-name AWSServiceRoleForLexBots
```

Output:

```
{
    "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteServiceLinkedRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(context.TODO(),
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
}
```

- Per i dettagli sull'API, DeleteServiceLinkedRoleconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { DeleteServiceLinkedRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const deleteServiceLinkedRole = (roleName) => {
  const command = new DeleteServiceLinkedRoleCommand({ RoleName: roleName });
  return client.send(command);
};
```

- Per i dettagli sull'API, DeleteServiceLinkedRoleconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio ha eliminato il ruolo collegato al servizio. Tieni presente che se il servizio utilizza ancora questo ruolo, questo comando genererà un errore.

```
Remove-IAMServiceLinkedRole -RoleName
 AWSServiceRoleForAutoScaling_RoleNameEndsWithThis
```

- Per i dettagli sull'API, vedere DeleteServiceLinkedRolein AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Deletes a service-linked role.
#
# @param role_name [String] The name of the role to delete.
def delete_service_linked_role(role_name)
  response = @iam_client.delete_service_linked_role(role_name: role_name)
  task_id = response.deletion_task_id
  check_deletion_status(role_name, task_id)
rescue Aws::Errors::ServiceError => e
  handle_deletion_error(e, role_name)
end

private

# Checks the deletion status of a service-linked role
#
# @param role_name [String] The name of the role being deleted
# @param task_id [String] The task ID for the deletion process
def check_deletion_status(role_name, task_id)
  loop do
    response = @iam_client.get_service_linked_role_deletion_status(
      deletion_task_id: task_id)
    status = response.status
    @logger.info("Deletion of #{role_name} #{status}.")
    break if %w[SUCCEEDED FAILED].include?(status)
    sleep(3)
  end
end

# Handles deletion error
#
# @param e [Aws::Errors::ServiceError] The error encountered during deletion
# @param role_name [String] The name of the role attempted to delete
```

```
  def handle_deletion_error(e, role_name)
    unless e.code == "NoSuchEntity"
      @logger.error("Couldn't delete #{role_name}. Here's why:")
      @logger.error("\t#{e.code}: #{e.message}")
      raise
    end
  end
```

- Per i dettagli sull'API, DeleteServiceLinkedRoleconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_service_linked_role(
    client: &iamClient,
    role_name: &str,
) -> Result<(), iamError> {
    client
        .delete_service_linked_role()
        .role_name(role_name)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sulle API, consulta il riferimento DeleteServiceLinkedRoleall'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteSigningCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteSigningCertificate`.

CLI

AWS CLI

Per eliminare un certificato di firma per un utente IAM

Il `delete-signing-certificate` comando seguente elimina il certificato di firma specificato per l'utente IAM denominatoBob.

```
aws iam delete-signing-certificate \
    --user-name Bob \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE
```

Questo comando non produce alcun output.

Per ottenere l'ID per un certificato di firma, usa il `list-signing-certificates` comando.

Per ulteriori informazioni, consulta Gestire i certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta DeleteSigningCertificate in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il certificato di firma con l'ID **`Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU`** dell'utente IAM denominato**`Bob`**.

```
Remove-IAMSigningCertificate -UserName Bob -CertificateId
 Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
```

- Per i dettagli sull'API, vedere DeleteSigningCertificate in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un gruppo e aggiunta di un utente
- Creazione di un utente e assunzione di un ruolo
- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
{ UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
```

```
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per i dettagli sull'API, consulta DeleteUser AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DeleteUserRequest request;
    request.SetUserName(userName);
    auto outcome = iam.DeleteUser(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error deleting IAM user " << userName << ": " <<
```

```
                outcome.GetError().GetMessage() << std::endl;;
    }
    else {
        std::cout << "Successfully deleted IAM user " << userName << std::endl;
    }

    return outcome.IsSuccess();
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Come eliminare un utente IAM

Il comando `delete-user` seguente rimuove l'utente IAM denominato Bob dall'account corrente.

```
aws iam delete-user \
    --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Eliminazione di un utente IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DeleteUser AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(userName string) error {
 _, err := wrapper.IamClient.DeleteUser(context.TODO(), &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.DeleteUserRequest;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <userName>\s

                Where:
                    userName - The name of the user to delete.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String userName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        deleteIAMUser(iam, userName);
        System.out.println("Done");
        iam.close();
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("Successfully deleted IAM user " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Eliminare l'utente.

```
import { DeleteUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} name
 */
export const deleteUser = (name) => {
  const command = new DeleteUserCommand({ UserName: name });
  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  UserName: process.argv[2],
};

iam.getUser(params, function (err, data) {
  if (err && err.code === "NoSuchEntity") {
    console.log("User " + process.argv[2] + " does not exist.");
  } else {
    iam.deleteUser(params, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Success", data);
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteIAMUser(userNameVal: String) {
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }

    // To delete a user, ensure that the user's access keys are deleted first.
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.deleteUser(request)
        println("Successfully deleted user $userNameVal")
    }
}
```

- Per i dettagli sull'API, DeleteUserconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina l'utente IAM denominato**Bob**.

```
Remove-IAMUser -UserName Bob
```

Esempio 2: questo esempio elimina l'utente IAM denominato **Theresa** insieme a tutti gli elementi che devono essere eliminati per primi.

```
$name = "Theresa"

# find any groups and remove user from them
```

```
$groups = Get-IAMGroupForUser -UserName $name
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName $name -Force }

# find any inline policies and delete them
$inlinepols = Get-IAMUserPolicies -UserName $name
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 $name -Force}

# find any managed polices and detach them
$managedpols = Get-IAMAttachedUserPolicies -UserName $name
foreach ($pol in $managedpols) { Unregister-IAMUserPolicy -PolicyArn
 $pol.PolicyArn -UserName $name }

# find any signing certificates and delete them
$certs = Get-IAMSigningCertificate -UserName $name
foreach ($cert in $certs) { Remove-IAMSigningCertificate -CertificateId
 $cert.CertificateId -UserName $name -Force }

# find any access keys and delete them
$keys = Get-IAMAccessKey -UserName $name
foreach ($key in $keys) { Remove-IAMAccessKey -AccessKeyId $key.AccessKeyId -
UserName $name -Force }

# delete the user's login profile, if one exists - note: need to use try/catch to
 suppress not found error
try { $prof = Get-IAMLoginProfile -UserName $name -ea 0 } catch { out-null }
if ($prof) { Remove-IAMLoginProfile -UserName $name -Force }

# find any MFA device, detach it, and if virtual, delete it.
$mfa = Get-IAMMFADevice -UserName $name
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}

# finally, remove the user
Remove-IAMUser -UserName $name -Force
```

- Per i dettagli sull'API, vedere DeleteUserin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def delete_user(user_name):
    """
    Deletes a user. Before a user can be deleted, all associated resources,
    such as access keys and policies, must be deleted or detached.

    :param user_name: The name of the user.
    """
    try:
        iam.User(user_name).delete()
        logger.info("Deleted user %s.", user_name)
    except ClientError:
        logger.exception("Couldn't delete user %s.", user_name)
        raise
```

- Per i dettagli sull'API, consulta DeleteUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  # Deletes a user and their associated resources
```

```
  #
  # @param user_name [String] The name of the user to delete
  def delete_user(user_name)
    user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
    user.each do |key|
      @iam_client.delete_access_key({ access_key_id: key.access_key_id,
 user_name: user_name })
      @logger.info("Deleted access key #{key.access_key_id} for user
 '#{user_name}'.")
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_user(client: &iamClient, user: &User) -> Result<(),
 SdkError<DeleteUserError>> {
    let user = user.clone();
    let mut tries: i32 = 0;
    let max_tries: i32 = 10;

    let response: Result<(), SdkError<DeleteUserError>> = loop {
        match client
            .delete_user()
            .user_name(user.user_name())
            .send()
            .await
```

```
        {
            Ok(_) => {
                break Ok(());
            }
            Err(e) => {
                tries += 1;
                if tries > max_tries {
                    break Err(e);
                }
                sleep(Duration::from_secs(2)).await;
            }
        }
    };

    response
}
```

- Per i dettagli sulle API, consulta la DeleteUserguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func deleteUser(user: IAMClientTypes.User) async throws {
        let input = DeleteUserInput(
            userName: user.userName
        )
```

```
        do {
            _ = try await iamClient.deleteUser(input: input)
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteUserguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteUserPermissionsBoundary`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUserPermissionsBoundary`.

CLI

AWS CLI

Per eliminare un limite di autorizzazioni da un utente IAM

L'`delete-user-permissions-boundary`esempio seguente elimina il limite delle autorizzazioni associato all'utente IAM denominato. `intern` Per applicare un limite di autorizzazioni a un utente, usa il comando. `put-user-permissions-boundary`

```
aws iam delete-user-permissions-boundary \
    --user-name intern
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command DeleteUserPermissionsBoundaryReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio mostra come rimuovere il limite di autorizzazione associato a un utente IAM.

```
Remove-IAMUserPermissionsBoundary -UserName joe
```

- Per i dettagli sull'API, vedere DeleteUserPermissionsBoundaryin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteUserPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un utente e assunzione di un ruolo

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an IAM user policy.
    /// </summary>
```

```
    /// <param name="policyName">The name of the IAM policy to delete.</param>
    /// <param name="userName">The username of the IAM user.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteUserPolicy in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come rimuovere una policy da un utente IAM

Il comando `delete-user-policy` seguente rimuove la policy specificata dall'utente IAM denominato Bob.

```
aws iam delete-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ottenere un elenco di policy per un utente IAM, usa il comando `list-user-policies`.

Per ulteriori informazioni, consulta Creare un utente IAM nel tuo AWS account nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta DeleteUserPolicy in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(userName string, policyName string)
 error {
 _, err := wrapper.IamClient.DeleteUserPolicy(context.TODO(),
 &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DeleteUserPolicy in AWS SDK for Go API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio elimina il nome della policy in linea **`AccessToEC2Policy`** che è incorporato nell'utente IAM denominato. **`Bob`**

```
Remove-IAMUserPolicy -PolicyName AccessToEC2Policy -UserName Bob
```

Esempio 2: Questo esempio trova tutte le policy in linea incorporate nell'utente IAM denominato **`Theresa`** e quindi le elimina.

```
$inlinepols = Get-IAMUserPolicies -UserName Theresa
foreach ($pol in $inlinepols) { Remove-IAMUserPolicy -PolicyName $pol -UserName
 Theresa -Force}
```

- Per i dettagli sull'API, vedere DeleteUserPolicy in AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Deletes a user and their associated resources
 #
 # @param user_name [String] The name of the user to delete
 def delete_user(user_name)
   user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
   user.each do |key|
     @iam_client.delete_access_key({ access_key_id: key.access_key_id,
user_name: user_name })
     @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
```

```
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
```

- Per i dettagli sull'API, consulta la DeleteUserPolicy in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn delete_user_policy(
    client: &iamClient,
    user: &User,
    policy_name: &str,
) -> Result<(), SdkError<DeleteUserPolicyError>> {
    client
        .delete_user_policy()
        .user_name(user.user_name())
        .policy_name(policy_name)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sull'API, consulta DeleteUserPolicy in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   func deleteUserPolicy(user: IAMClientTypes.User, policyName: String) async
throws {
        let input = DeleteUserPolicyInput(
            policyName: policyName,
            userName: user.userName
        )
        do {
            _ = try await iamClient.deleteUserPolicy(input: input)
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta DeleteUserPolicy in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteVirtualMfaDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteVirtualMfaDevice`.

CLI

AWS CLI

Per rimuovere un dispositivo MFA virtuale

Il `delete-virtual-mfa-device` comando seguente rimuove il dispositivo MFA specificato dall'account corrente.

```
aws iam delete-virtual-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Disattivazione dei dispositivi MFA nella Guida per l'AWS utente IAM.

- Per i dettagli sull'API, consulta Command DeleteVirtualMfaDeviceReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il dispositivo MFA virtuale IAM il cui ARN è. **`arn:aws:iam::123456789012:mfa/bob`**

```
Remove-IAMVirtualMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/bob
```

Esempio 2: questo esempio verifica se all'utente IAM Theresa è assegnato un dispositivo MFA. Se ne viene trovato uno, il dispositivo viene disabilitato per l'utente IAM. Se il dispositivo è virtuale, viene anche eliminato.

```
$mfa = Get-IAMMFADevice -UserName Theresa
if ($mfa) {
    Disable-IAMMFADevice -SerialNumber $mfa.SerialNumber -UserName $name
    if ($mfa.SerialNumber -like "arn:*") { Remove-IAMVirtualMFADevice -
SerialNumber $mfa.SerialNumber }
}
```

- Per i dettagli sull'API, vedere DeleteVirtualMfaDevicein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DetachGroupPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DetachGroupPolicy`.

CLI

AWS CLI

Per scollegare una politica da un gruppo

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy` dal gruppo chiamato. `Testers`

```
aws iam detach-group-policy \
    --group-name Testers \
    --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta DetachGroupPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio separa la politica di gruppo gestita il cui ARN **`arn:aws:iam::123456789012:policy/TesterAccessPolicy`** proviene dal gruppo denominato. **`Testers`**

```
Unregister-IAMGroupPolicy -GroupName Testers -PolicyArn
 arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Esempio 2: Questo esempio trova tutte le politiche gestite allegate al gruppo denominato **`Testers`** e le rimuove dal gruppo.

```
Get-IAMAttachedGroupPolicies -GroupName Testers | Unregister-IAMGroupPolicy -
Groupname Testers
```

- Per i dettagli sull'API, vedere DetachGroupPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DetachRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DetachRolePolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione di un utente e assunzione di un ruolo
- Gestione dei ruoli

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Detach an IAM policy from an IAM role.
   /// </summary>
   /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
   /// <param name="roleName">The name of the IAM role.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
```

```
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
```

```
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
```

```
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::DetachRolePolicyRequest detachRequest;
    detachRequest.SetRoleName(roleName);
    detachRequest.SetPolicyArn(policyArn);

    auto detachOutcome = iam.DetachRolePolicy(detachRequest);
```

```
    if (!detachOutcome.IsSuccess()) {
        std::cerr << "Failed to detach policy " << policyArn << " from role "
                  << roleName << ": " << detachOutcome.GetError().GetMessage() <<
                  std::endl;
    }
    else {
        std::cout << "Successfully detached policy " << policyArn << " from role
 "
                  << roleName << std::endl;
    }

    return detachOutcome.IsSuccess();
```

- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come scollegare una policy da un ruolo

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` dal ruolo denominato `FedTesterRole`.

```
aws iam detach-role-policy \
    --role-name FedTesterRole \
    --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(roleName string, policyArn string)
 error {
 _, err := wrapper.IamClient.DetachRolePolicy(context.TODO(),
 &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}
```

- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.DetachRolePolicyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetachRolePolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleName> <policyArn>\s

                Where:
                    roleName - A role name that you can obtain from the AWS
 Management Console.\s
                    policyArn - A policy ARN that you can obtain from the AWS
 Management Console.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String roleName = args[0];
        String policyArn = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();
        detachPolicy(iam, roleName, policyArn);
        System.out.println("Done");
        iam.close();
    }

    public static void detachPolicy(IamClient iam, String roleName, String
 policyArn) {
        try {
            DetachRolePolicyRequest request = DetachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.detachRolePolicy(request);
            System.out.println("Successfully detached policy " + policyArn +
                    " from role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Scollega la policy.

```
import { DetachRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 * @param {string} roleName
 */
export const detachRolePolicy = (policyArn, roleName) => {
  const command = new DetachRolePolicyCommand({
    PolicyArn: policyArn,
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var paramsRoleList = {
  RoleName: process.argv[2],
};

iam.listAttachedRolePolicies(paramsRoleList, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var myRolePolicies = data.AttachedPolicies;
    myRolePolicies.forEach(function (val, index, array) {
      if (myRolePolicies[index].PolicyName === "AmazonDynamoDBFullAccess") {
        var params = {
          PolicyArn: "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess",
          RoleName: process.argv[2],
        };
        iam.detachRolePolicy(params, function (err, data) {
          if (err) {
            console.log("Unable to detach policy from role", err);
          } else {
            console.log("Policy detached from role successfully");
            process.exit();
          }
        });
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DetachRolePolicy in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun detachPolicy(
    roleNameVal: String,
    policyArnVal: String
) {
    val request =
        DetachRolePolicyRequest {
            roleName = roleNameVal
            policyArn = policyArnVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.detachRolePolicy(request)
        println("Successfully detached policy $policyArnVal from role
 $roleNameVal")
    }
}
```

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio separa la politica di gruppo gestita il cui ARN **`arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy`** proviene dal ruolo denominato. **`FedTesterRole`**

```
Unregister-IAMRolePolicy -RoleName FedTesterRole -PolicyArn
 arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Esempio 2: questo esempio trova tutte le politiche gestite associate al ruolo denominato **`FedTesterRole`** e le stacca dal ruolo.

```
Get-IAMAttachedRolePolicyList -RoleName FedTesterRole | Unregister-IAMRolePolicy
 -Rolename FedTesterRole
```

- Per i dettagli sull'API, vedere DetachRolePolicy in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Scollega una policy da un ruolo utilizzando l'oggetto Policy Boto3.

```
def detach_from_role(role_name, policy_arn):
    """
    Detaches a policy from a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Policy(policy_arn).detach_role(RoleName=role_name)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
        )
        raise
```

Scollega una policy da un ruolo utilizzando l'oggetto Role Boto3.

```
def detach_policy(role_name, policy_arn):
    """
    Detaches a policy from a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
        )
        raise
```

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, allega e scollega le politiche relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
```

```
 # @param iam_client [Aws::IAM::Client] An initialized IAM client
 def initialize(iam_client, logger: Logger.new($stdout))
   @iam_client = iam_client
   @logger = logger
   @logger.progname = "PolicyManager"
 end

 # Creates a policy
 #
 # @param policy_name [String] The name of the policy
 # @param policy_document [Hash] The policy document
 # @return [String] The policy ARN if successful, otherwise nil
 def create_policy(policy_name, policy_document)
   response = @iam_client.create_policy(
     policy_name: policy_name,
     policy_document: policy_document.to_json
   )
   response.policy.arn
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error creating policy: #{e.message}")
   nil
 end

 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end

 # Attaches a policy to a role
 #
```

```
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sulle API, consulta DetachRolePolicy in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn detach_role_policy(
    client: &iamClient,
    role_name: &str,
    policy_arn: &str,
) -> Result<(), iamError> {
    client
        .detach_role_policy()
        .role_name(role_name)
        .policy_arn(policy_arn)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public func detachRolePolicy(policy: IAMClientTypes.Policy, role:
IAMClientTypes.Role) async throws {
       let input = DetachRolePolicyInput(
           policyArn: policy.arn,
           roleName: role.roleName
       )

       do {
           _ = try await iamClient.detachRolePolicy(input: input)
       } catch {
           throw error
       }
   }
```

- Per i dettagli sull'API, consulta DetachRolePolicy in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DetachUserPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DetachUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

CLI

AWS CLI

Come scollegare una policy da un utente

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/TesterPolicy` dall'utente Bob.

```
aws iam detach-user-policy \
    --user-name Bob \
    --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta DetachUserPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio scollega la policy gestita il cui ARN **`arn:aws:iam::123456789012:policy/TesterPolicy`** proviene dall'utente IAM denominato. **`Bob`**

```
Unregister-IAMUserPolicy -UserName Bob -PolicyArn
 arn:aws:iam::123456789012:policy/TesterPolicy
```

Esempio 2: questo esempio trova tutte le policy gestite allegate all'utente IAM denominato **`Theresa`** e le separa dall'utente.

```
Get-IAMAttachedUserPolicyList -UserName Theresa | Unregister-IAMUserPolicy -
Username Theresa
```

- Per i dettagli sull'API, vedere DetachUserPolicy in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def detach_policy(user_name, policy_arn):
    """
    Detaches a policy from a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from user %s.", policy_arn, user_name
        )
        raise
```

- Per i dettagli sull'API, consulta DetachUserPolicy in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  # Detaches a policy from a user
  #
  # @param user_name [String] The name of the user
  # @param policy_arn [String] The ARN of the policy to detach
  # @return [Boolean] true if the policy was successfully detached, false
otherwise
  def detach_user_policy(user_name, policy_arn)
    @iam_client.detach_user_policy(
      user_name: user_name,
      policy_arn: policy_arn
    )
    @logger.info("Policy '#{policy_arn}' detached from user '#{user_name}'
successfully.")
    true
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Error detaching policy: Policy or user does not exist.")
    false
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from user '#{user_name}':
#{e.message}")
    false
  end
```

- Per i dettagli sull'API, consulta la DetachUserPolicy in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn detach_user_policy(
    client: &iamClient,
    user_name: &str,
    policy_arn: &str,
) -> Result<(), iamError> {
```

```
    client
        .detach_user_policy()
        .user_name(user_name)
        .policy_arn(policy_arn)
        .send()
        .await?;

    Ok(())
}
```

- Per i dettagli sull'API, consulta DetachUserPolicy in AWS SDK for Rust API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`EnableMfaDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`EnableMfaDevice`.

CLI

AWS CLI

Per abilitare un dispositivo MFA

Dopo aver utilizzato il `create-virtual-mfa-device` comando per creare un nuovo dispositivo MFA virtuale, è possibile assegnare il dispositivo MFA a un utente. L'`enable-mfa-device`esempio seguente assegna il dispositivo MFA con il `arn:aws:iam::210987654321:mfa/BobsMFADevice` numero di serie all'utente. `Bob` Il comando sincronizza inoltre il dispositivo con AWS l'inclusione dei primi due codici in sequenza dal dispositivo MFA virtuale.

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale nella Guida per AWS l'utente IAM.

- Per i dettagli sull'API, consulta EnableMfaDevice in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando abilita il dispositivo hardware MFA con il numero di serie **987654321098** e lo associa all'utente. **Bob** Include i primi due codici in sequenza provenienti dal dispositivo.

```
Enable-IAMMFADevice -UserName "Bob" -SerialNumber "987654321098" -
AuthenticationCode1 "12345678" -AuthenticationCode2 "87654321"
```

Esempio 2: Questo esempio crea e abilita un dispositivo MFA virtuale. Il primo comando crea il dispositivo virtuale e restituisce la rappresentazione dell'oggetto del dispositivo nella variabile**$MFADevice**. È possibile utilizzare le **QRCodePng** proprietà **.Base32StringSeed** o per configurare l'applicazione software dell'utente. Il comando finale assegna il dispositivo all'utente**David**, identificandolo in base al numero di serie. Il comando sincronizza inoltre il dispositivo con AWS l'inclusione dei primi due codici in sequenza dal dispositivo MFA virtuale.

```
$MFADevice = New-IAMVirtualMFADevice -VirtualMFADeviceName "MyMFADevice"
# see example for New-IAMVirtualMFADevice to see how to configure the software
 program with PNG or base32 seed code
Enable-IAMMFADevice -UserName "David" -SerialNumber -SerialNumber
 $MFADevice.SerialNumber -AuthenticationCode1 "24681357" -AuthenticationCode2
 "13572468"
```

- Per i dettagli sull'API, vedere EnableMfaDevice in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GenerateCredentialReport** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateCredentialReport`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come generare un report delle credenziali

L'esempio seguente tenta di generare un rapporto sulle credenziali per l' AWS account.

```
aws iam generate-credential-report
```

Output:

```
{
    "State":  "STARTED",
    "Description": "No report exists. Starting a new report generation task"
}
```

Per ulteriori informazioni, consulta Ottenere i report sulle credenziali per il tuo AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GenerateCredentialReport in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio richiede la generazione di un nuovo report, operazione che può essere eseguita ogni quattro ore. Se l'ultimo rapporto è ancora recente, viene visualizzato il campo State. **`COMPLETE`** Utilizzare **`Get-IAMCredentialReport`** per visualizzare il rapporto completato.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                    State
-----------                                                    -----
No report exists. Starting a new report generation task        STARTED
```

- Per i dettagli sull'API, vedere GenerateCredentialReport in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def generate_credential_report():
    """
    Starts generation of a credentials report about the current account. After
    calling this function to generate the report, call get_credential_report
    to get the latest report. A new report can be generated a minimum of four
 hours
    after the last one was generated.
    """
    try:
        response = iam.meta.client.generate_credential_report()
        logger.info(
            "Generating credentials report for your account. " "Current state is
 %s.",
            response["State"],
        )
    except ClientError:
        logger.exception("Couldn't generate a credentials report for your
 account.")
        raise
    else:
        return response
```

- Per i dettagli sull'API, consulta GenerateCredentialReport in AWS SDK for Python (Boto3) API Report in SDK for Python (Boto3).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GenerateServiceLastAccessedDetails`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateServiceLastAccessedDetails`.

CLI

AWS CLI

Esempio 1: generare un rapporto di accesso al servizio per una politica personalizzata

L'`generate-service-last-accessed-details`esempio seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono gli utenti IAM e altre entità con una politica personalizzata denominata`intern-boundary`. È possibile visualizzare il report dopo averlo creato eseguendo il `get-service-last-accessed-details` comando.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::123456789012:policy/intern-boundary
```

Output:

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

Esempio 2: generare un rapporto di accesso al servizio per la AdministratorAccess politica AWS gestita

L'`generate-service-last-accessed-details`esempio seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono gli utenti IAM e altre

entità con la `AdministratorAccess` policy AWS gestita. È possibile visualizzare il report dopo averlo creato eseguendo il `get-service-last-accessed-details` comando.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::aws:policy/AdministratorAccess
```

Output:

```
{
    "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

Per ulteriori informazioni, consulta Ridefinizione delle autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta i GenerateServiceLastAccesseddettagli in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio è un cmdlet equivalente all'API. GenerateServiceLastAccessedDetails Ciò fornisce un job id che può essere utilizzato in Get-IAM e ServiceLastAccessedDetail Get-IAM ServiceLast AccessedDetail WithEntity

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

- Per i dettagli sull'API, vedere GenerateServiceLastAccessedDettagli in Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetAccessKeyLastUsed`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccessKeyLastUsed`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::accessKeyLastUsed(const Aws::String &secretKeyID,
                                    const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetAccessKeyLastUsedRequest request;

    request.SetAccessKeyId(secretKeyID);

    Aws::IAM::Model::GetAccessKeyLastUsedOutcome outcome =
 iam.GetAccessKeyLastUsed(
            request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error querying last used time for access key " <<
                  secretKeyID << ":" << outcome.GetError().GetMessage() <<
 std::endl;
    }
    else {
        Aws::String lastUsedTimeString =
                outcome.GetResult()
                        .GetAccessKeyLastUsed()
                        .GetLastUsedDate()
                        .ToGmtString(Aws::Utils::DateFormat::ISO_8601);
        std::cout << "Access key " << secretKeyID << " last used at time " <<
                  lastUsedTimeString << std::endl;
    }
```

```
    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta GetAccessKeyLastUsed in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come recuperare informazioni sull'ultimo utilizzo della chiave di accesso specificata

L'esempio seguente recupera informazioni sull'ultimo utilizzo della chiave di accesso ABCDEXAMPLE.

```
aws iam get-access-key-last-used \
    --access-key-id ABCDEXAMPLE
```

Output:

```
{
    "UserName":  "Bob",
    "AccessKeyLastUsed": {
        "Region": "us-east-1",
        "ServiceName": "iam",
        "LastUsedDate": "2015-06-16T22:45:00Z"
    }
}
```

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccessKeyLastUsed in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la chiave di accesso.

```
import { GetAccessKeyLastUsedCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} accessKeyId
 */
export const getAccessKeyLastUsed = async (accessKeyId) => {
  const command = new GetAccessKeyLastUsedCommand({
    AccessKeyId: accessKeyId,
  });

  const response = await client.send(command);

  if (response.AccessKeyLastUsed?.LastUsedDate) {
    console.log(`
    ${accessKeyId} was last used by ${response.UserName} via
    the ${response.AccessKeyLastUsed.ServiceName} service on
    ${response.AccessKeyLastUsed.LastUsedDate.toISOString()}
    `);
  }

  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta GetAccessKeyLastUsed in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.getAccessKeyLastUsed(
  { AccessKeyId: "ACCESS_KEY_ID" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data.AccessKeyLastUsed);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta GetAccessKeyLastUsed in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce il nome utente proprietario e le informazioni sull'ultimo utilizzo della chiave di accesso fornita.

```
Get-IAMAccessKeyLastUsed -AccessKeyId ABCDEXAMPLE
```

- Per i dettagli sull'API, vedere GetAccessKeyLastUsed in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_last_use(key_id):
    """
    Gets information about when and how a key was last used.

    :param key_id: The ID of the key to look up.
    :return: Information about the key's last use.
    """
    try:
        response = iam.meta.client.get_access_key_last_used(AccessKeyId=key_id)
        last_used_date = response["AccessKeyLastUsed"].get("LastUsedDate", None)
        last_service = response["AccessKeyLastUsed"].get("ServiceName", None)
        logger.info(
            "Key %s was last used by %s on %s to access %s.",
            key_id,
            response["UserName"],
            last_used_date,
            last_service,
        )
    except ClientError:
        logger.exception("Couldn't get last use of key %s.", key_id)
        raise
    else:
        return response
```

- Per i dettagli sull'API, consulta GetAccessKeyLastUsed in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetAccountAuthorizationDetails`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountAuthorizationDetails`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Per elencare un AWS account, utenti, gruppi, ruoli e politiche IAM

Il `get-account-authorization-details` comando seguente restituisce informazioni su tutti gli utenti, i gruppi, i ruoli e le politiche IAM presenti nell' AWS account.

```
aws iam get-account-authorization-details
```

Output:

```
{
    "RoleDetailList": [
        {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
```

```
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "RoleId": "AROA1234567890EXAMPLE",
        "CreateDate": "2014-07-30T17:09:20Z",
        "InstanceProfileList": [
            {
                "InstanceProfileId": "AIPA1234567890EXAMPLE",
                "Roles": [
                    {
                        "AssumeRolePolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Sid": "",
                                    "Effect": "Allow",
                                    "Principal": {
                                        "Service": "ec2.amazonaws.com"
                                    },
                                    "Action": "sts:AssumeRole"
                                }
                            ]
                        },
                        "RoleId": "AROA1234567890EXAMPLE",
                        "CreateDate": "2014-07-30T17:09:20Z",
                        "RoleName": "EC2role",
                        "Path": "/",
                        "Arn": "arn:aws:iam::123456789012:role/EC2role"
                    }
                ],
                "CreateDate": "2014-07-30T17:09:20Z",
                "InstanceProfileName": "EC2role",
                "Path": "/",
                "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
            }
        ],
        "RoleName": "EC2role",
        "Path": "/",
        "AttachedManagedPolicies": [
            {
                "PolicyName": "AmazonS3FullAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
            },
```

```
                {
                    "PolicyName": "AmazonDynamoDBFullAccess",
                    "PolicyArn": "arn:aws:iam::aws:policy/
AmazonDynamoDBFullAccess"
                }
            ],
            "RoleLastUsed": {
                "Region": "us-west-2",
                "LastUsedDate": "2019-11-13T17:30:00Z"
            },
            "RolePolicyList": [],
            "Arn": "arn:aws:iam::123456789012:role/EC2role"
        }
    ],
    "GroupDetailList": [
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": {
                "PolicyName": "AdministratorAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
            },
            "GroupName": "Admins",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "CreateDate": "2013-10-14T18:32:24Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": {
                "PolicyName": "PowerUserAccess",
                "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
            },
            "GroupName": "Dev",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:group/Dev",
            "CreateDate": "2013-10-14T18:33:55Z",
            "GroupPolicyList": []
        },
        {
            "GroupId": "AIDA1234567890EXAMPLE",
            "AttachedManagedPolicies": [],
            "GroupName": "Finance",
            "Path": "/",
```

```
            "Arn": "arn:aws:iam::123456789012:group/Finance",
            "CreateDate": "2013-10-14T18:57:48Z",
            "GroupPolicyList": [
                {
                    "PolicyName": "policygen-201310141157",
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": "aws-portal:*",
                                "Sid": "Stmt1381777017000",
                                "Resource": "*",
                                "Effect": "Allow"
                            }
                        ]
                    }
                }
            ]
        }
    ],
    "UserDetailList": [
        {
            "UserName": "Alice",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:24Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [],
            "Path": "/",
            "AttachedManagedPolicies": [],
            "Arn": "arn:aws:iam::123456789012:user/Alice"
        },
        {
            "UserName": "Bob",
            "GroupList": [
                "Admins"
            ],
            "CreateDate": "2013-10-14T18:32:25Z",
            "UserId": "AIDA1234567890EXAMPLE",
            "UserPolicyList": [
                {
                    "PolicyName": "DenyBillingAndIAMPolicy",
                    "PolicyDocument": {
```

```
                    "Version": "2012-10-17",
                    "Statement": {
                        "Effect": "Deny",
                        "Action": [
                            "aws-portal:*",
                            "iam:*"
                        ],
                        "Resource": "*"
                    }
                }
            }
        ],
        "Path": "/",
        "AttachedManagedPolicies": [],
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    },
    {
        "UserName": "Charlie",
        "GroupList": [
            "Dev"
        ],
        "CreateDate": "2013-10-14T18:33:56Z",
        "UserId": "AIDA1234567890EXAMPLE",
        "UserPolicyList": [],
        "Path": "/",
        "AttachedManagedPolicies": [],
        "Arn": "arn:aws:iam::123456789012:user/Charlie"
    }
],
"Policies": [
    {
        "PolicyName": "create-update-delete-set-managed-policies",
        "CreateDate": "2015-02-06T19:58:34Z",
        "AttachmentCount": 1,
        "IsAttachable": true,
        "PolicyId": "ANPA1234567890EXAMPLE",
        "DefaultVersionId": "v1",
        "PolicyVersionList": [
            {
                "CreateDate": "2015-02-06T19:58:34Z",
                "VersionId": "v1",
                "Document": {
                    "Version": "2012-10-17",
                    "Statement": {
```

```
                            "Effect": "Allow",
                            "Action": [
                                "iam:CreatePolicy",
                                "iam:CreatePolicyVersion",
                                "iam:DeletePolicy",
                                "iam:DeletePolicyVersion",
                                "iam:GetPolicy",
                                "iam:GetPolicyVersion",
                                "iam:ListPolicies",
                                "iam:ListPolicyVersions",
                                "iam:SetDefaultPolicyVersion"
                            ],
                            "Resource": "*"
                        }
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
            "UpdateDate": "2015-02-06T19:58:34Z"
        },
        {
            "PolicyName": "S3-read-only-specific-bucket",
            "CreateDate": "2015-01-21T21:39:41Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2015-01-21T21:39:41Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": [
                                    "s3:Get*",
                                    "s3:List*"
                                ],
                                "Resource": [
```

```
                                    "arn:aws:s3:::example-bucket",
                                    "arn:aws:s3:::example-bucket/*"
                                ]
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-
bucket",
            "UpdateDate": "2015-01-21T23:39:41Z"
        },
        {
            "PolicyName": "AmazonEC2FullAccess",
            "CreateDate": "2015-02-06T18:40:15Z",
            "AttachmentCount": 1,
            "IsAttachable": true,
            "PolicyId": "ANPA1234567890EXAMPLE",
            "DefaultVersionId": "v1",
            "PolicyVersionList": [
                {
                    "CreateDate": "2014-10-30T20:59:46Z",
                    "VersionId": "v1",
                    "Document": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": "ec2:*",
                                "Effect": "Allow",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "elasticloadbalancing:*",
                                "Resource": "*"
                            },
                            {
                                "Effect": "Allow",
                                "Action": "cloudwatch:*",
                                "Resource": "*"
                            },
                            {
```

```
                                "Effect": "Allow",
                                "Action": "autoscaling:*",
                                "Resource": "*"
                            }
                        ]
                    },
                    "IsDefaultVersion": true
                }
            ],
            "Path": "/",
            "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
            "UpdateDate": "2015-02-06T18:40:15Z"
        }
    ],
    "Marker": "EXAMPLEkakv9BCuUNFDtxWSyfzetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}
```

Per ulteriori informazioni, consulta Linee guida sugli audit di sicurezza AWS nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccountAuthorizationDetails AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio ottiene i dettagli di autorizzazione sulle identità nell' AWS account e visualizza l'elenco degli elementi dell'oggetto restituito, inclusi utenti, gruppi e ruoli. Ad esempio, la **`UserDetailList`** proprietà visualizza i dettagli sugli utenti. Informazioni simili sono disponibili nelle **`GroupDetailList`** proprietà **`RoleDetailList`** e.

```
$Details=Get-IAMAccountAuthorizationDetail
$Details
```

Output:

```
GroupDetailList : {Administrators, Developers, Testers, Backup}
IsTruncated     : False
```

```
Marker          :
RoleDetailList  : {TestRole1, AdminRole, TesterRole, clirole...}
UserDetailList  : {Administrator, Bob, BackupToS3, }
```

```
$Details.UserDetailList
```

Output:

```
Arn             : arn:aws:iam::123456789012:user/Administrator
CreateDate      : 10/16/2014 9:03:09 AM
GroupList       : {Administrators}
Path            : /
UserId          : AIDACKCEVSQ6CEXAMPLE1
UserName        : Administrator
UserPolicyList : {}

Arn             : arn:aws:iam::123456789012:user/Bob
CreateDate      : 4/6/2015 12:54:42 PM
GroupList       : {Developers}
Path            : /
UserId          : AIDACKCEVSQ6CEXAMPLE2
UserName        : bab
UserPolicyList : {}

Arn             : arn:aws:iam::123456789012:user/BackupToS3
CreateDate      : 1/27/2015 10:15:08 AM
GroupList       : {Backup}
Path            : /
UserId          : AIDACKCEVSQ6CEXAMPLE3
UserName        : BackupToS3
UserPolicyList : {BackupServicePermissionsToS3Buckets}
```

- Per i dettagli sull'API, vedere GetAccountAuthorizationDetailsin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_authorization_details(response_filter):
    """
    Gets an authorization detail report for the current account.

    :param response_filter: A list of resource types to include in the report,
 such
                            as users or roles. When not specified, all resources
                            are included.
    :return: The authorization detail report.
    """
    try:
        account_details = iam.meta.client.get_account_authorization_details(
            Filter=response_filter
        )
        logger.debug(account_details)
    except ClientError:
        logger.exception("Couldn't get details for your account.")
        raise
    else:
        return account_details
```

- Per i dettagli sull'API, consulta GetAccountAuthorizationDetails AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetAccountPasswordPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountPasswordPolicy`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }
```

- Per i dettagli sull'API, GetAccountPasswordPolicyconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Come visualizzare la policy delle password dell'account corrente

Il comando `get-account-password-policy` seguente visualizza dettagli sulla policy delle password per l'account corrente.

```
aws iam get-account-password-policy
```

Output:

```
{
    "PasswordPolicy": {
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

Se non è definita alcuna policy delle password per l'account, il comando restituisce un errore `NoSuchEntity`.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetAccountPasswordPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}
```

```
// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy() (*types.PasswordPolicy,
 error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(context.TODO(),
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}
```

- Per i dettagli sull'API, GetAccountPasswordPolicyconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la policy sulla password dell'account.

```
import {
  GetAccountPasswordPolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const getAccountPasswordPolicy = async () => {
  const command = new GetAccountPasswordPolicyCommand({});
```

```
  const response = await client.send(command);
  console.log(response.PasswordPolicy);
  return response;
};
```

- Per i dettagli sull'API, GetAccountPasswordPolicyconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getAccountPasswordPolicy()
    {
        return $this->iamClient->getAccountPasswordPolicy();
    }
```

- Per i dettagli sull'API, GetAccountPasswordPolicyconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce dettagli sulla politica delle password per l'account corrente. Se non è definita alcuna politica in materia di password per l'account, il comando restituisce un **`NoSuchEntity`** errore.

```
Get-IAMAccountPasswordPolicy
```

Output:

```
AllowUsersToChangePassword : True
ExpirePasswords            : True
HardExpiry                 : False
MaxPasswordAge             : 90
MinimumPasswordLength      : 8
PasswordReusePrevention    : 20
RequireLowercaseCharacters : True
RequireNumbers             : True
RequireSymbols             : False
RequireUppercaseCharacters : True
```

- Per i dettagli sull'API, vedere GetAccountPasswordPolicyin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def print_password_policy():
    """
    Prints the password policy for the account.
    """
    try:
        pw_policy = iam.AccountPasswordPolicy()
        print("Current account password policy:")
        print(
            f"\tallow_users_to_change_password:
 {pw_policy.allow_users_to_change_password}"
        )
        print(f"\texpire_passwords: {pw_policy.expire_passwords}")
```

```
        print(f"\thard_expiry: {pw_policy.hard_expiry}")
        print(f"\tmax_password_age: {pw_policy.max_password_age}")
        print(f"\tminimum_password_length: {pw_policy.minimum_password_length}")
        print(f"\tpassword_reuse_prevention:
{pw_policy.password_reuse_prevention}")
        print(
            f"\trequire_lowercase_characters:
{pw_policy.require_lowercase_characters}"
        )
        print(f"\trequire_numbers: {pw_policy.require_numbers}")
        print(f"\trequire_symbols: {pw_policy.require_symbols}")
        print(
            f"\trequire_uppercase_characters:
{pw_policy.require_uppercase_characters}"
        )
        printed = True
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchEntity":
            print("The account does not have a password policy set.")
        else:
            logger.exception("Couldn't get account password policy.")
            raise
    else:
        return printed
```

- Per i dettagli sull'API, consulta GetAccountPasswordPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Class to manage IAM account password policies
class PasswordPolicyManager
  attr_accessor :iam_client, :logger

  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "IAMPolicyManager"
  end

  # Retrieves and logs the account password policy
  def print_account_password_policy
    begin
      response = @iam_client.get_account_password_policy
      @logger.info("The account password policy is:
 #{response.password_policy.to_h}")
    rescue Aws::IAM::Errors::NoSuchEntity
      @logger.info("The account does not have a password policy.")
    rescue Aws::Errors::ServiceError => e
      @logger.error("Couldn't print the account password policy. Error: #{e.code}
 - #{e.message}")
      raise
    end
  end
end
```

- Per i dettagli sull'API, GetAccountPasswordPolicyconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn get_account_password_policy(
```

```
    client: &iamClient,
) -> Result<GetAccountPasswordPolicyOutput,
 SdkError<GetAccountPasswordPolicyError>> {
    let response = client.get_account_password_policy().send().await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento GetAccountPasswordPolicyall'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetAccountSummary`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetAccountSummary`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come ottenere informazioni sull'utilizzo delle entità IAM e sulle quote IAM nell'account corrente

Il comando `get-account-summary` seguente restituisce informazioni sull'utilizzo corrente delle entità IAM e sulle quote correnti delle entità IAM nell'account.

```
aws iam get-account-summary
```

Output:

```
{
    "SummaryMap": {
        "UsersQuota": 5000,
```

```
        "GroupsQuota": 100,
        "InstanceProfiles": 6,
        "SigningCertificatesPerUserQuota": 2,
        "AccountAccessKeysPresent": 0,
        "RolesQuota": 250,
        "RolePolicySizeQuota": 10240,
        "AccountSigningCertificatesPresent": 0,
        "Users": 27,
        "ServerCertificatesQuota": 20,
        "ServerCertificates": 0,
        "AssumeRolePolicySizeQuota": 2048,
        "Groups": 7,
        "MFADevicesInUse": 1,
        "Roles": 3,
        "AccountMFAEnabled": 1,
        "MFADevices": 3,
        "GroupsPerUserQuota": 10,
        "GroupPolicySizeQuota": 5120,
        "InstanceProfilesQuota": 100,
        "AccessKeysPerUserQuota": 2,
        "Providers": 0,
        "UserPolicySizeQuota": 2048
    }
}
```

Per ulteriori informazioni sulle limitazioni delle entità, consulta le quote IAM e AWS STS nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetAccountSummary in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce informazioni sull'utilizzo corrente delle entità IAM e sulle quote correnti delle entità IAM in. Account AWS

```
Get-IAMAccountSummary
```

Output:

```
Key                                     Value
Users                                   7
GroupPolicySizeQuota                    5120
PolicyVersionsInUseQuota                10000
ServerCertificatesQuota                 20
AccountSigningCertificatesPresent       0
AccountAccessKeysPresent                0
Groups                                  3
UsersQuota                              5000
RolePolicySizeQuota                     10240
UserPolicySizeQuota                     2048
GroupsPerUserQuota                      10
AssumeRolePolicySizeQuota               2048
AttachedPoliciesPerGroupQuota           2
Roles                                   9
VersionsPerPolicyQuota                  5
GroupsQuota                             100
PolicySizeQuota                         5120
Policies                                5
RolesQuota                              250
ServerCertificates                      0
AttachedPoliciesPerRoleQuota            2
MFADevicesInUse                         2
PoliciesQuota                           1000
AccountMFAEnabled                       1
Providers                               2
InstanceProfilesQuota                   100
MFADevices                              4
AccessKeysPerUserQuota                  2
AttachedPoliciesPerUserQuota            2
SigningCertificatesPerUserQuota         2
PolicyVersionsInUse                     4
InstanceProfiles                        1
...
```

- Per i dettagli sull'API, vedere GetAccountSummary in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_summary():
    """
    Gets a summary of account usage.

    :return: The summary of account usage.
    """
    try:
        summary = iam.AccountSummary()
        logger.debug(summary.summary_map)
    except ClientError:
        logger.exception("Couldn't get a summary for your account.")
        raise
    else:
        return summary.summary_map
```

- Per i dettagli sull'API, consulta GetAccountSummary in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetContextKeysForCustomPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetContextKeysForCustomPolicy`.

CLI

AWS CLI

Esempio 1: per elencare le chiavi di contesto a cui fanno riferimento una o più politiche JSON personalizzate fornite come parametro nella riga di comando

Il `get-context-keys-for-custom-policy` comando seguente analizza ogni politica fornita ed elenca le chiavi di contesto utilizzate da tali politiche. Utilizzate questo comando per identificare quali valori chiave di contesto è necessario fornire per utilizzare correttamente i comandi `simulate-custom-policy` del simulatore di politiche e. `simulate-custom-policy` Puoi anche recuperare l'elenco delle chiavi di contesto utilizzate da tutte le policy associate da un utente o ruolo IAM utilizzando il `get-context-keys-for-custom-policy` comando. I valori dei parametri che iniziano con `file://` indicano al comando di leggere il file e di utilizzare il contenuto come valore per il parametro anziché il nome del file stesso.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Esempio 2: elencare le chiavi di contesto a cui fanno riferimento una o più politiche JSON personalizzate fornite come input di file

Il `get-context-keys-for-custom-policy` comando seguente è lo stesso dell'esempio precedente, tranne per il fatto che le politiche vengono fornite in un file anziché come parametro. Poiché il comando prevede un elenco di stringhe JSON e non un elenco di strutture JSON, il file deve essere strutturato come segue, sebbene sia possibile comprimerlo in un unico file.

```
[
    "Policy1",
    "Policy2"
]
```

Quindi, ad esempio, un file che contiene la politica dell'esempio precedente deve avere l'aspetto seguente. È necessario evitare ogni virgoletta doppia incorporata nella stringa di policy facendola precedere da una barra rovesciata».

```
[ "{\"Version\": \"2012-10-17\", \"Statement\": {\"Effect\": \"Allow
\", \"Action\": \"dynamodb:*\", \"Resource\": \"arn:aws:dynamodb:us-
west-2:128716708097:table/${aws:username}\", \"Condition\": {\"DateGreaterThan\":
 {\"aws:CurrentTime\": \"2015-08-16T12:00:00Z\"}}}}" ]
```

Questo file può quindi essere inviato al seguente comando.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list file://policyfile.json
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Per ulteriori informazioni, consulta Using the IAM Policy Simulator (AWS CLI AWS e API) nella IAM User AWS Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetContextKeysForCustomPolicyReference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera tutte le chiavi di contesto presenti nella policy JSON fornita. Per fornire più politiche è possibile fornire un elenco di valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForCustomPolicy -PolicyInputList $policy1,$policy2
```

- Per i dettagli sull'API, vedere GetContextKeysForCustomPolicyin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetContextKeysForPrincipalPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetContextKeysForPrincipalPolicy`.

CLI

AWS CLI

Per elencare le chiavi di contesto a cui fanno riferimento tutte le policy associate a un principale IAM

Il `get-context-keys-for-principal-policy` comando seguente recupera tutte le politiche allegate all'utente `saanvi` e ai gruppi di cui è membro. Quindi analizza ciascuna di esse ed elenca le chiavi di contesto utilizzate da tali politiche. Utilizzate questo comando per identificare i valori delle chiavi di contesto che dovete fornire per utilizzare correttamente `simulate-principal-policy` i comandi `simulate-custom-policy` and. È inoltre possibile recuperare l'elenco delle chiavi di contesto utilizzate da una politica JSON arbitraria utilizzando il comando. `get-context-keys-for-custom-policy`

```
aws iam get-context-keys-for-principal-policy \
   --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Output:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

Per ulteriori informazioni, consulta Using the IAM Policy Simulator (AWS CLI AWS e API) nella IAM User AWS Guide.

- Per i dettagli sull'API, consulta AWS CLI Command GetContextKeysForPrincipalPolicyReference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera tutte le chiavi di contesto presenti nella policy json fornita e le politiche allegate all'entità IAM (user/role ecc.). Per: PolicyInputList puoi fornire più elenchi di valori come valori separati da virgole.

```
$policy1 = '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'
$policy2 = '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/"}}'
Get-IAMContextKeysForPrincipalPolicy -PolicyInputList $policy1,$policy2 -
PolicySourceArn arn:aws:iam::852640994763:user/TestUser
```

- Per i dettagli sull'API, vedere GetContextKeysForPrincipalPolicyin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetCredentialReport`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetCredentialReport`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

CLI

AWS CLI

Come ottenere un report delle credenziali

Questo esempio apre il report restituito e lo invia alla pipeline come array di righe di testo.

```
aws iam get-credential-report
```

Output:

```
{
    "GeneratedTime":  "2015-06-17T19:11:50Z",
    "ReportFormat": "text/csv"
}
```

Per ulteriori informazioni, consulta Ottenere i report sulle credenziali per il tuo AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetCredentialReport in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio apre il report restituito e lo invia alla pipeline come matrice di righe di testo. La prima riga è l'intestazione con i nomi delle colonne separati da virgole. Ogni riga successiva è la riga di dettaglio per un utente, con ogni campo separato da virgole. Prima di poter visualizzare il report, è necessario generarlo con il **`Request-IAMCredentialReport`** cmdlet. Per recuperare il report come singola stringa, utilizzare

-Raw invece di. **-AsTextArray** L'alias **-SplitLines** è accettato anche per lo **-AsTextArray** switch. Per l'elenco completo delle colonne dell'output, consulta il riferimento all'API del servizio. Nota che se non utilizzi **-AsTextArray** or**-SplitLines**, devi estrarre il testo dalla **.Content** proprietà utilizzando la **StreamReader** classe.NET.

```
Request-IAMCredentialReport
```

Output:

```
Description                                                    State
-----------                                                    -----
No report exists. Starting a new report generation task        STARTED
```

```
Get-IAMCredentialReport -AsTextArray
```

Output:

```
 user,arn,user_creation_time,password_enabled,password_last_used,password_last_changed,pa
 root_account,arn:aws:iam::123456789012:root,2014-10-15T16:31:25+00:00,not_supported,2015
A,false,N/A,false,N/A,false,N/A
Administrator,arn:aws:iam::123456789012:user/
Administrator,2014-10-16T16:03:09+00:00,true,2015-04-20T15:18:32+00:00,2014-10-16T16:06:0
A,false,true,2014-12-03T18:53:41+00:00,true,2015-03-25T20:38:14+00:00,false,N/
A,false,N/A
Bill,arn:aws:iam::123456789012:user/Bill,2015-04-15T18:27:44+00:00,false,N/A,N/
A,N/A,false,false,N/A,false,N/A,false,2015-04-20T20:00:12+00:00,false,N/A
```

- Per i dettagli sull'API, vedere GetCredentialReport in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_credential_report():
    """
    Gets the most recently generated credentials report about the current
 account.

    :return: The credentials report.
    """
    try:
        response = iam.meta.client.get_credential_report()
        logger.debug(response["Content"])
    except ClientError:
        logger.exception("Couldn't get credentials report.")
        raise
    else:
        return response["Content"]
```

- Per i dettagli sull'API, consulta GetCredentialReport in AWS SDK for Python (Boto3) API Report in SDK for Python (Boto3).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetGroup`.

CLI

AWS CLI

Per creare un gruppo IAM

Questo esempio restituisce dettagli sul gruppo IAM`Admins`.

```
aws iam get-group \
    --group-name Admins
```

Output:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-06-16T19:41:48Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

Per ulteriori informazioni, consulta IAM Identities (users, user groups and roles) nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce dettagli sul gruppo IAM**`Testers`**, inclusa una raccolta di tutti gli utenti IAM che appartengono al gruppo.

```
$results = Get-IAMGroup -GroupName "Testers"
$results
```

Output:

```
Group                                    IsTruncated           Marker
     Users
-----                                    -----------           ------
     -----
Amazon.IdentityManagement.Model.Group    False
     {Theresa, David}
```

```
$results.Group
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Testers
```

```
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : 3RHNZZGQJ7QHMAEXAMPLE1
GroupName  : Testers
Path       : /
```

```
$results.Users
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Theresa
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : 4OSVDDJJTF4XEEXAMPLE2
UserName         : Theresa

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere GetGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetGroupPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetGroupPolicy`.

CLI

AWS CLI

Per ottenere informazioni su una policy allegata a un gruppo IAM

Il `get-group-policy` comando seguente ottiene informazioni sulla politica specificata allegata al gruppo denominato`Test-Group`.

```
aws iam get-group-policy \
    --group-name Test-Group \
    --policy-name S3-ReadOnly-Policy
```

Output:

```
{
    "GroupName": "Test-Group",
    "PolicyDocument": {
        "Statement": [
            {
                "Action": [
                    "s3:Get*",
                    "s3:List*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "PolicyName": "S3-ReadOnly-Policy"
}
```

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, vedere GetGroupPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce i dettagli sulla politica in linea incorporata denominata **`PowerUserAccess-Testers`** per il gruppo**`Testers`**. La **`PolicyDocument`** proprietà è codificata in URL. In questo esempio viene decodificata con il **`UrlDecode`** metodo.NET.

```
$results = Get-IAMGroupPolicy -GroupName Testers -PolicyName PowerUserAccess-
Testers
$results
```

Output:

```
GroupName     PolicyDocument
 PolicyName
---------     --------------
 ----------
Testers       %7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%0A%20...
 PowerUserAccess-Testers

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "NotAction": "iam:*",
      "Resource": "*"
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetGroupPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetInstanceProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetInstanceProfile`.

CLI

AWS CLI

Per ottenere informazioni sul profilo di un'istanza

Il `get-instance-profile` comando seguente ottiene informazioni sul profilo di istanza denominato`ExampleInstanceProfile`.

```
aws iam get-instance-profile \
```

```
    --instance-profile-name ExampleInstanceProfile
```

Output:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",
        "Roles": [
            {
                "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                "RoleId": "AIDGPMS9RO4H3FEXAMPLE",
                "CreateDate": "2013-01-09T06:33:26Z",
                "RoleName": "Test-Role",
                "Path": "/",
                "Arn": "arn:aws:iam::336924118301:role/Test-Role"
            }
        ],
        "CreateDate": "2013-06-12T23:52:02Z",
        "InstanceProfileName": "ExampleInstanceProfile",
        "Path": "/",
        "Arn": "arn:aws:iam::336924118301:instance-profile/
ExampleInstanceProfile"
    }
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetInstanceProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce i dettagli del profilo **`ec2instancerole`** di istanza denominato definito nell' AWS account corrente.

```
Get-IAMInstanceProfile -InstanceProfileName ec2instancerole
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
```

```
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere GetInstanceProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetLoginProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetLoginProfile`.

CLI

AWS CLI

Per ottenere informazioni sulla password per un utente IAM

Il `get-login-profile` comando seguente ottiene informazioni sulla password per l'utente IAM denominatoBob.

```
aws iam get-login-profile \
    --user-name Bob
```

Output:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2012-09-21T23:03:39Z"
    }
}
```

Il `get-login-profile` comando può essere utilizzato per verificare che un utente IAM disponga di una password. Il comando restituisce un `NoSuchEntity` errore se non è definita alcuna password per l'utente.

Non è possibile visualizzare una password utilizzando questo comando. Se la password viene persa, è possibile reimpostare la password (`update-login-profile`) per l'utente. In alternativa, è possibile eliminare il profilo di accesso (`delete-login-profile`) per l'utente e quindi crearne uno nuovo (`create-login-profile`).

Per ulteriori informazioni, consulta Managing password for IAM users nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta GetLoginProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce la data di creazione della password e se è necessaria una reimpostazione della password per l'utente IAM**`David`**.

```
Get-IAMLoginProfile -UserName David
```

Output:

```
CreateDate                 PasswordResetRequired              UserName
----------                 ---------------------              --------
12/10/2014 3:39:44 PM      False                              David
```

- Per i dettagli sull'API, vedere GetLoginProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetOpenIdConnectProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetOpenIdConnectProvider`.

CLI

AWS CLI

Per restituire informazioni sul provider OpenID Connect specificato

Questo esempio restituisce dettagli sul provider OpenID Connect il cui ARN è. `arn:aws:iam::123456789012:oidc-provider/server.example.com`

```
aws iam get-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Output:

```
{
    "Url": "server.example.com"
        "CreateDate": "2015-06-16T19:41:48Z",
        "ThumbprintList": [
        "12345abcdefghijk67890lmnopqrst987example"
        ],
        "ClientIDList": [
        "example-application-ID"
        ]
}
```

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta GetOpenIdConnectProvider in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce i dettagli sul provider OpenID Connect il cui ARN è. **`arn:aws:iam::123456789012:oidc-provider/accounts.google.com`** La **`ClientIDList`** proprietà è una raccolta che contiene tutti gli ID client definiti per questo provider.

```
Get-IAMOpenIDConnectProvider -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/oidc.example.com
```

Output:

```
ClientIDList         CreateDate                ThumbprintList
           Url
------------         ----------                --------------
           ---
{MyOIDCApp}          2/3/2015 3:00:30 PM
 {12345abcdefghijk67890lmnopqrst98765uvwxy}   oidc.example.com
```

- Per i dettagli sull'API, vedere GetOpenIdConnectProvider in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Lavora con l'API IAM Policy Builder

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
```

```
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
{ PolicyArn = policyArn });
        return response.Policy;
    }
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::getPolicy(const Aws::String &policyArn,
                            const Aws::Client::ClientConfiguration &clientConfig)
 {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetPolicyRequest request;
    request.SetPolicyArn(policyArn);

    auto outcome = iam.GetPolicy(request);
    if (!outcome.IsSuccess()) {
        std::cerr << "Error getting policy " << policyArn << ": " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        const auto &policy = outcome.GetResult().GetPolicy();
        std::cout << "Name: " << policy.GetPolicyName() << std::endl <<
                  "ID: " << policy.GetPolicyId() << std::endl << "Arn: " <<
                  policy.GetArn() << std::endl << "Description: " <<
                  policy.GetDescription() << std::endl << "CreateDate: " <<

 policy.GetCreateDate().ToGmtString(Aws::Utils::DateFormat::ISO_8601)
```

```
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Come recuperare informazioni sulla policy gestita specificata

Questo esempio restituisce i dettagli sulla policy gestita il cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{
    "Policy": {
        "PolicyName": "MySamplePolicy",
        "CreateDate": "2015-06-17T19:23;32Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "Z27SI6FQMGNQ2EXAMPLE1",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
        "UpdateDate": "2015-06-17T19:23:32Z"
    }
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetPolicy AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// GetPolicy gets data about a policy.
func (wrapper PolicyWrapper) GetPolicy(policyArn string) (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(context.TODO(), &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la policy.

```
import { GetPolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} policyArn
 */
export const getPolicy = (policyArn) => {
  const command = new GetPolicyCommand({
    PolicyArn: policyArn,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
```

```
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  PolicyArn: "arn:aws:iam::aws:policy/AWSLambdaExecute",
};

iam.getPolicy(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Policy.Description);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getIAMPolicy(policyArnVal: String?) {
    val request =
        GetPolicyRequest {
            policyArn = policyArnVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.getPolicy(request)
```

```
        println("Successfully retrieved policy ${response.policy?.policyName}")
    }
}
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getPolicy($policyArn)
    {
        return $this->customWaiter(function () use ($policyArn) {
            return $this->iamClient->getPolicy(['PolicyArn' => $policyArn]);
        });
    }
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce i dettagli sulla politica gestita il cui ARN è. **`arn:aws:iam::123456789012:policy/MySamplePolicy`**

```
Get-IAMPolicy -PolicyArn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
Arn              : arn:aws:iam::aws:policy/MySamplePolicy
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : MySamplePolicy
UpdateDate       : 2/6/2015 10:40:08 AM
```

- Per i dettagli sull'API, vedere GetPolicyin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.

    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
```

```
    else:
        return policy_statement
```

- Per i dettagli sull'API, consulta GetPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Fetches an IAM policy by its ARN
 # @param policy_arn [String] the ARN of the IAM policy to retrieve
 # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
 def get_policy(policy_arn)
   response = @iam_client.get_policy(policy_arn: policy_arn)
   policy = response.policy
   @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
   policy
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
   raise
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
   raise
 end
```

- Per i dettagli sull'API, GetPolicyconsulta AWS SDK for Ruby API Reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func getPolicy(arn: String) async throws -> IAMClientTypes.Policy {
    let input = GetPolicyInput(
        policyArn: arn
    )
    do {
        let output = try await client.getPolicy(input: input)
        guard let policy = output.policy else {
            throw ServiceHandlerError.noSuchPolicy
        }
        return policy
    } catch {
        throw error
    }
}
```

- Per i dettagli sull'API, consulta la GetPolicyguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetPolicyVersion`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetPolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Lavora con l'API IAM Policy Builder

CLI

AWS CLI

Come recuperare informazioni sulla versione specificata della policy gestita specificata

Questo esempio restituisce il documento della policy per la versione v2 della policy il cui ARN è `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Output:

```
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:*",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2023-04-11T00:22:54+00:00"
    }
```

```
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetPolicyVersion in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce il documento relativo alla politica per la **v2** versione della politica il cui ARN è. **arn:aws:iam::123456789012:policy/MyManagedPolicy** Il documento di policy contenuto nella **Document** proprietà è codificato in URL e viene decodificato in questo esempio con il **UrlDecode** metodo.NET.

```
$results = Get-IAMPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy -VersionId v2
$results
```

Output:

```
CreateDate             Document
 IsDefaultVersion     VersionId
----------             --------
 ----------------     ---------
2/12/2015 9:39:53 AM   %7B%0A%20%20%22Version%22%3A%20%222012-10...    True
           v2

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
$policy = [System.Web.HttpUtility]::UrlDecode($results.Document)
$policy
{
  "Version": "2012-10-17",
  "Statement":
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
}
```

- Per i dettagli sull'API, vedere GetPolicyVersion in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.

    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
    else:
        return policy_statement
```

- Per i dettagli sull'API, consulta GetPolicyVersion in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetRole`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Get information about an IAM role.
/// </summary>
/// <param name="roleName">The name of the IAM role to retrieve information
/// for.</param>
/// <returns>The IAM role that was retrieved.</returns>
public async Task<Role> GetRoleAsync(string roleName)
{
    var response = await _IAMService.GetRoleAsync(new GetRoleRequest
    {
        RoleName = roleName,
    });

    return response.Role;
}
```

- Per i dettagli sull'API, consulta la GetRolesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Come ottenere informazioni su un ruolo IAM

Il comando `get-role` seguente ottiene informazioni sul ruolo denominato `Test-Role`.

```
aws iam get-role \
    --role-name Test-Role
```

Output:

```
{
    "Role": {
        "Description": "Test Role",
        "AssumeRolePolicyDocument":"<URL-encoded-JSON>",
        "MaxSessionDuration": 3600,
        "RoleId": "AROA1234567890EXAMPLE",
        "CreateDate": "2019-11-13T16:45:56Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "RoleLastUsed": {
            "Region": "us-east-1",
            "LastUsedDate": "2019-11-13T17:14:00Z"
        },
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
}
```

Il comando visualizza la policy di attendibilità associata al ruolo. Per elencare le policy di autorizzazioni collegate a un ruolo, usa il comando `list-role-policies`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetRole AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// GetRole gets data about a role.
func (wrapper RoleWrapper) GetRole(roleName string) (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.GetRole(context.TODO(),
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}
```

- Per i dettagli sull'API, consulta la GetRolesezione AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera il ruolo.

```
import { GetRoleCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 */
export const getRole = (roleName) => {
  const command = new GetRoleCommand({
    RoleName: roleName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la GetRolesezione AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function getRole($roleName)
    {
        return $this->customWaiter(function () use ($roleName) {
            return $this->iamClient->getRole(['RoleName' => $roleName]);
        });
    }
```

- Per i dettagli sull'API, consulta la GetRolesezione AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce i dettagli di**lamda_exec_role**. Include il documento sulla politica di fiducia che specifica chi può assumere questo ruolo. Il documento di policy è codificato tramite URL e può essere decodificato utilizzando il metodo.NET. **UrlDecode** In questo esempio, la policy originale aveva rimosso tutti gli spazi bianchi prima di essere caricata nella policy. Per visualizzare i documenti relativi alle politiche in materia di autorizzazioni che determinano cosa può fare qualcuno che assume il ruolo, utilizza i documenti per le politiche in linea e **Get-IAMRolePolicy** **Get-IAMPolicyVersion** per le politiche gestite allegate.

```
$results = Get-IamRole -RoleName lambda_exec_role
$results | Format-List
```

Output:

```
Arn                       : arn:aws:iam::123456789012:role/lambda_exec_role
AssumeRolePolicyDocument  : %7B%22Version%22%3A%222012-10-17%22%2C%22Statement
%22%3A%5B%7B%22Sid%22
                            %3A%22%22%2C%22Effect%22%3A%22Allow%22%2C%22Principal
%22%3A%7B%22Service
                            %22%3A%22lambda.amazonaws.com%22%7D%2C%22Action%22%3A
%22sts%3AAssumeRole
                            %22%7D%5D%7D
CreateDate                : 4/2/2015 9:16:11 AM
```

```
Path                  : /
RoleId                : 2YBIKAIBHNKB4EXAMPLE1
RoleName              : lambda_exec_role
```

```
$policy = [System.Web.HttpUtility]::UrlDecode($results.AssumeRolePolicyDocument)
$policy
```

Output:

```
{"Version":"2012-10-17","Statement":[{"Sid":"","Effect":"Allow","Principal":
{"Service":"lambda.amazonaws.com"},"Action":"sts:AssumeRole"}]}
```

- Per i dettagli sull'API, vedere GetRolein AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def get_role(role_name):
    """
    Gets a role by name.

    :param role_name: The name of the role to retrieve.
    :return: The specified role.
    """
    try:
        role = iam.Role(role_name)
        role.load()  # calls GetRole to load attributes
        logger.info("Got role with arn %s.", role.arn)
    except ClientError:
        logger.exception("Couldn't get role named %s.", role_name)
        raise
    else:
        return role
```

- Per i dettagli sull'API, consulta GetRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Gets data about a role.
#
# @param name [String] The name of the role to look up.
# @return [Aws::IAM::Role] The retrieved role.
def get_role(name)
  role = @iam_client.get_role({
                               role_name: name,
                             }).role
  puts("Got data for role '#{role.role_name}'. Its ARN is '#{role.arn}'.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't get data for role '#{name}' Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
end
```

- Per i dettagli sull'API, consulta la GetRolesezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn get_role(
    client: &iamClient,
    role_name: String,
) -> Result<GetRoleOutput, SdkError<GetRoleError>> {
    let response = client.get_role().role_name(role_name).send().await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta la GetRoleguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func getRole(name: String) async throws -> IAMClientTypes.Role {
        let input = GetRoleInput(
```

```
            roleName: name
        )
        do {
            let output = try await client.getRole(input: input)
            guard let role = output.role else {
                throw ServiceHandlerError.noSuchRole
            }
            return role
        } catch {
            throw error
        }
    }
```

- Per i dettagli sull'API, consulta la GetRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetRolePolicy`.

CLI

AWS CLI

Per ottenere informazioni su una policy associata a un ruolo IAM

Il `get-role-policy` comando seguente ottiene informazioni sulla politica specificata allegata al ruolo denominato`Test-Role`.

```
aws iam get-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

Output:

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
```

```
      "Statement": [
          {
              "Action": [
                  "s3:ListBucket",
                  "s3:Put*",
                  "s3:Get*",
                  "s3:*MultipartUpload*"
              ],
              "Resource": "*",
              "Effect": "Allow",
              "Sid": "1"
          }
      ]
  }
  "PolicyName": "ExamplePolicy"
}
```

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, vedere GetRolePolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce il documento sulla politica delle autorizzazioni per la policy denominata **`oneClick_lambda_exec_role_policy`** che è incorporata nel ruolo **`lamda_exec_role`** IAM. Il documento di policy risultante è codificato in URL. In questo esempio viene decodificato con il **`UrlDecode`** metodo.NET.

```
$results = Get-IAMRolePolicy -RoleName lambda_exec_role -PolicyName
 oneClick_lambda_exec_role_policy
$results
```

Output:

```
PolicyDocument                                          PolicyName
              UserName
--------------                                          ----------
              --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...
 oneClick_lambda_exec_role_policy     lambda_exec_role
```

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
```

Output:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:*"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetRolePolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetSamlProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetSamlProvider`.

CLI

AWS CLI

Per recuperare il metadocumento del provider SAML

Questo esempio recupera i dettagli sul provider SAML 2.0 il cui ARM è. `arn:aws:iam::123456789012:saml-provider/SAMLADFS` La risposta include il documento di metadati che hai ricevuto dal provider di identità per creare l'entità del provider AWS SAML, nonché le date di creazione e scadenza.

```
aws iam get-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Output:

```
{
    "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
    "CreateDate": "2017-03-06T22:29:46+00:00",
    "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetSamlProvider in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio recupera i dettagli sul provider SAML 2.0 il cui ARM è arn:aws:iam: :123456789012:SAML-provider/SAMLadfs. La risposta include il documento di metadati che hai ricevuto dal provider di identità per creare l'entità del provider SAML, nonché le date di creazione e scadenza. AWS

```
Get-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS
```

Output:

```
CreateDate                 SAMLMetadataDocument
        ValidUntil
----------                 --------------------
        ----------
12/23/2014 12:16:55 PM    <EntityDescriptor ID="_12345678-1234-5678-9012-
example1...    12/23/2114 12:16:54 PM
```

- Per i dettagli sull'API, vedere GetSamlProvider in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetServerCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::getServerCertificate(const Aws::String &certificateName,
                                       const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::GetServerCertificateRequest request;
    request.SetServerCertificateName(certificateName);

    auto outcome = iam.GetServerCertificate(request);
    bool result = true;
    if (!outcome.IsSuccess()) {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error getting server certificate " << certificateName
 <<
                      ": " << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << certificateName
                      << "' not found." << std::endl;
        }
    }
    else {
        const auto &certificate = outcome.GetResult().GetServerCertificate();
        std::cout << "Name: " <<

 certificate.GetServerCertificateMetadata().GetServerCertificateName()
                  << std::endl << "Body: " << certificate.GetCertificateBody() <<
                  std::endl << "Chain: " << certificate.GetCertificateChain() <<
                  std::endl;
    }

    return result;
}
```

- Per i dettagli sull'API, consulta GetServerCertificate in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per ottenere dettagli su un certificato server nel tuo AWS account

Il `get-server-certificate` comando seguente recupera tutti i dettagli sul certificato server specificato nel tuo AWS account.

```
aws iam get-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

Output:

```
{
    "ServerCertificate": {
        "ServerCertificateMetadata": {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        "CertificateBody": "-----BEGIN CERTIFICATE-----
            MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
            VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
            b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
            BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
            MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
            VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
            b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
            YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
            21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
            rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
            Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
            nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
            FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
            NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
        "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
            7oRw0uXOjANBgkqhkiG9w0BAqQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
            AldBMRAwDgYDVQQHEwdTZWF0drGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
```

```
            TC0lBTSBDb25zb2xlMRIwEAYDVsQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQ
            jb20wHhcNMTEwNDI1MjA0NTIxWhtcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
            MCVVMxCzAJBgNVBAgTAldBMRAwDgsYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBb
            WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMx
            HzAdBgkqhkiG9w0BCQEWEG5vb25lQGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
            BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
            k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
            ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
            AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
            KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
            EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
            3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
    }
}
```

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetServerCertificate in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un certificato del server.

```
import { GetServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} certName
```

```
 * @returns
 */
export const getServerCertificate = async (certName) => {
  const command = new GetServerCertificateCommand({
    ServerCertificateName: certName,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta GetServerCertificate in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.getServerCertificate(
  { ServerCertificateName: "CERTIFICATE_NAME" },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
```

```
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta GetServerCertificate in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera i dettagli sul certificato del server denominato**`MyServerCertificate`**. È possibile trovare i dettagli del certificato nelle **`ServerCertificateMetadata`** proprietà **`CertificateBody`** e.

```
$result = Get-IAMServerCertificate -ServerCertificateName MyServerCertificate
$result | format-list
```

Output:

```
CertificateBody           : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt

 YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
                             21uUSfwfEvySWtC2XADZ4nB
+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
                             rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
```

```
 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                            nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                            NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                            -----END CERTIFICATE-----
CertificateChain          :
ServerCertificateMetadata :
 Amazon.IdentityManagement.Model.ServerCertificateMetadata
```

```
$result.ServerCertificateMetadata
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere GetServerCertificate in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetServiceLastAccessedDetails`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLastAccessedDetails`.

CLI

AWS CLI

Per recuperare un rapporto di accesso al servizio

L'`get-service-last-accessed-details`esempio seguente recupera un report generato in precedenza che elenca i servizi a cui accedono le entità IAM. Per generare un report, usa il `generate-service-last-accessed-details` comando.

```
aws iam get-service-last-accessed-details \
    --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

Output:

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:50:35.929Z",
    "ServicesLastAccessed": [
        ...
        {
            "ServiceName": "AWS Lambda",
            "LastAuthenticated": "2019-09-30T23:02:00Z",
            "ServiceNamespace": "lambda",
            "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",
            "TotalAuthenticatedEntities": 6
        },
    ]
}
```

Per ulteriori informazioni, consulta Ridefinizione delle autorizzazioni nell' AWS utilizzo delle ultime informazioni a cui si accede nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta i GetServiceLastAccesseddettagli in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio fornisce i dettagli dell'ultimo accesso al servizio da parte dell'entità IAM (utente, gruppo, ruolo o policy) associata alla chiamata Request.

```
Request-IAMServiceLastAccessedDetail -Arn arn:aws:iam::123456789012:user/TestUser
```

Output:

```
f0b7a819-eab0-929b-dc26-ca598911cb9f
```

```
Get-IAMServiceLastAccessedDetail -JobId f0b7a819-eab0-929b-dc26-ca598911cb9f
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDettagli in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetServiceLastAccessedDetailsWithEntities`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLastAccessedDetailsWithEntities`.

CLI

AWS CLI

Per recuperare un rapporto di accesso al servizio con i dettagli relativi a un servizio

L'`get-service-last-accessed-details-with-entities`esempio seguente recupera un report che contiene dettagli sugli utenti IAM e altre entità che hanno avuto accesso al servizio specificato. Per generare un report, usa il `generate-service-last-accessed-details` comando. Per ottenere un elenco di servizi a cui si accede con i namespace, usa. `get-service-last-accessed-details`

```
aws iam get-service-last-accessed-details-with-entities \
    --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
    --service-namespace lambda
```

Output:

```
{
    "JobStatus": "COMPLETED",
    "JobCreationDate": "2019-10-01T03:55:41.756Z",
    "JobCompletionDate": "2019-10-01T03:55:42.533Z",
```

```
    "EntityDetailsList": [
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/admin",
                "Name": "admin",
                "Type": "USER",
                "Id": "AIDAIO2XMPLENQEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-30T23:02:00Z"
        },
        {
            "EntityInfo": {
                "Arn": "arn:aws:iam::123456789012:user/developer",
                "Name": "developer",
                "Type": "USER",
                "Id": "AIDAIBEYXMPL2YEXAMPLE",
                "Path": "/"
            },
            "LastAuthenticated": "2019-09-16T19:34:00Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Raffinamento delle autorizzazioni nell' AWS utilizzo delle informazioni dell'ultimo accesso nella Guida per l'utente IAM.AWS

- Per i dettagli sull'API, consulta GetServiceLastAccessedDetailsWithEntities in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio fornisce il timestamp dell'ultimo accesso per il servizio contenuto nella richiesta della rispettiva entità IAM.

```
$results = Get-IAMServiceLastAccessedDetailWithEntity -JobId f0b7a819-eab0-929b-
dc26-ca598911cb9f -ServiceNamespace ec2
$results
```

Output:

```
EntityDetailsList : {Amazon.IdentityManagement.Model.EntityDetails}
Error             :
IsTruncated       : False
JobCompletionDate : 12/29/19 11:19:31 AM
JobCreationDate   : 12/29/19 11:19:31 AM
JobStatus         : COMPLETED
Marker            :
```

```
$results.EntityDetailsList
```

Output:

```
EntityInfo                                    LastAuthenticated
----------                                    -----------------
Amazon.IdentityManagement.Model.EntityInfo 11/16/19 3:47:00 PM
```

```
$results.EntityInfo
```

Output:

```
Arn  : arn:aws:iam::123456789012:user/TestUser
Id   : AIDA4NBK5CXF5TZHU1234
Name : TestUser
Path : /
Type : USER
```

- Per i dettagli sull'API, vedere GetServiceLastAccessedDetailsWithEntities in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetServiceLinkedRoleDeletionStatus`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetServiceLinkedRoleDeletionStatus`.

CLI

AWS CLI

Come verificare lo stato di una richiesta di eliminazione di un ruolo collegato a un servizio

L'esempio `get-service-linked-role-deletion-status` seguente visualizza lo stato di una precedente richiesta di eliminazione di un ruolo collegato a un servizio. L'operazione di eliminazione avviene in modo asincrono. Quando effettui la richiesta, ottieni un valore `DeletionTaskId` che hai fornito come parametro per questo comando.

```
aws iam get-service-linked-role-deletion-status \
    --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Output:

```
{
"Status": "SUCCEEDED"
}
```

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati a servizi nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetServiceLinkedRoleDeletionStatus AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import {
  GetServiceLinkedRoleDeletionStatusCommand,
```

```
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} deletionTaskId
 */
export const getServiceLinkedRoleDeletionStatus = (deletionTaskId) => {
  const command = new GetServiceLinkedRoleDeletionStatusCommand({
    DeletionTaskId: deletionTaskId,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, GetServiceLinkedRoleDeletionStatusconsulta AWS SDK for JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetUser`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about an IAM user.
```

```
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
 { UserName = userName });
        return response.User;
    }
```

- Per i dettagli sull'API, GetUserconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
```

```
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
###############################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}
```

- Per i dettagli sull'API, consulta GetUser AWS CLICommand Reference.

CLI

AWS CLI

Come ottenere informazioni su un utente IAM

Il comando `get-user` seguente ottiene informazioni sull'utente IAM denominato `Paulo`.

```
aws iam get-user \
    --user-name Paulo
```

Output:

```
{
    "User": {
        "UserName": "Paulo",
        "Path": "/",
        "CreateDate": "2019-09-21T23:03:13Z",
        "UserId": "AIDA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Paulo"
    }
}
```

Per ulteriori informazioni, consulta Gestione di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta GetUser AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(userName string) (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.GetUser(context.TODO(), &iam.GetUserInput{
  UserName: aws.String(userName),
 })
```

```
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
  user = result.User
 }
 return user, err
}
```

- Per i dettagli sull'API, GetUserconsulta AWS SDK for Go API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera i dettagli sull'utente denominato**David**.

```
Get-IAMUser -UserName David
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE1
UserName         : David
```

Esempio 2: Questo esempio recupera i dettagli sull'utente IAM attualmente connesso.

```
Get-IAMUser
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE2
UserName         : Bob
```

- Per i dettagli sull'API, vedere GetUserin Cmdlet Reference.AWS Tools for PowerShell

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Retrieves a user's details
 #
 # @param user_name [String] The name of the user to retrieve
 # @return [Aws::IAM::Types::User, nil] The user object if found, or nil if an
error occurred
 def get_user(user_name)
   response = @iam_client.get_user(user_name: user_name)
   response.user
 rescue Aws::IAM::Errors::NoSuchEntity
   @logger.error("User '#{user_name}' not found.")
   nil
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Error retrieving user '#{user_name}': #{e.message}")
   nil
 end
```

- Per i dettagli sull'API, GetUserconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetUserPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetUserPolicy`.

CLI

AWS CLI

Per elencare i dettagli delle policy per un utente IAM

Il `get-user-policy` comando seguente elenca i dettagli della politica specificata allegata all'utente IAM denominatoBob.

```
aws iam get-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

Output:

```
{
    "UserName": "Bob",
    "PolicyName": "ExamplePolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
}
```

Per ottenere un elenco di policy per un utente IAM, usa il comando `list-user-policies`.

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta GetUserPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera i dettagli della policy in linea denominata **`Davids_IAM_Admin_Policy`** incorporata nell'utente IAM denominato. **`David`** Il documento di policy è codificato in URL.

```
$results = Get-IAMUserPolicy -PolicyName Davids_IAM_Admin_Policy -UserName David
$results
```

Output:

```
PolicyDocument                                                  PolicyName
       UserName
--------------                                                  ----------
       --------
%7B%0A%20%20%22Version%22%3A%20%222012-10-17%22%2C%...    Davids_IAM_Admin_Policy
       David

[System.Reflection.Assembly]::LoadWithPartialName("System.Web.HttpUtility")
[System.Web.HttpUtility]::UrlDecode($results.PolicyDocument)
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

- Per i dettagli sull'API, vedere GetUserPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListAccessKeys`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAccessKeys`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function iam_list_access_keys
#
# This function lists the access keys for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#
# Returns:
```

```
#       access_key_ids
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_list_access_keys() {

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_access_keys"
    echo "Lists the AWS Identity and Access Management (IAM) access key IDs for
 the specified user."
    echo "  -u user_name   The name of the IAM user."
    echo ""
  }

  local user_name response
  local option OPTARG # Required to use getopts command in a function.
  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam list-access-keys \
    --user-name "$user_name" \
    --output text \
```

```
    --query 'AccessKeyMetadata[].AccessKeyId')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-access-keys operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listAccessKeys(const Aws::String &userName,
                                 const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListAccessKeysRequest request;
    request.SetUserName(userName);

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListAccessKeys(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list access keys for user " << userName
                      << ": " << outcome.GetError().GetMessage() << std::endl;
```

```
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(32) << "UserName" <<
                      std::setw(30) << "KeyID" << std::setw(20) << "Status" <<
                      std::setw(20) << "CreateDate" << std::endl;
            header = true;
        }

        const auto &keys = outcome.GetResult().GetAccessKeyMetadata();
        const Aws::String DATE_FORMAT = "%Y-%m-%d";

        for (const auto &key: keys) {
            Aws::String statusString =
                    Aws::IAM::Model::StatusTypeMapper::GetNameForStatusType(
                            key.GetStatus());
            std::cout << std::left << std::setw(32) << key.GetUserName() <<
                      std::setw(30) << key.GetAccessKeyId() << std::setw(20) <<
                      statusString << std::setw(20) <<
                      key.GetCreateDate().ToGmtString(DATE_FORMAT.c_str()) <<
 std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come elencare gli ID delle chiavi di accesso per un utente IAM

Il comando `list-access-keys` seguente elenca gli ID delle chiavi di accesso per l'utente IAM denominato Bob.

```
aws iam list-access-keys \
    --user-name Bob
```

Output:

```
{
    "AccessKeyMetadata": [
        {
            "UserName": "Bob",
            "Status": "Active",
            "CreateDate": "2013-06-04T18:17:34Z",
            "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CreateDate": "2013-06-06T20:42:26Z",
            "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"
        }
    ]
}
```

Non puoi elencare le chiavi di accesso segrete per gli utenti IAM. Se le chiavi di accesso segrete vengono perse, devi creare nuove chiavi di accesso utilizzando il comando `create-access-keys`.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAccessKeys in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(context.TODO(),
 &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.AccessKeyMetadata;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListAccessKeysRequest;
import software.amazon.awssdk.services.iam.model.ListAccessKeysResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListAccessKeys {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <userName>\s

                Where:
                    userName - The name of the user for which access keys are
 retrieved.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String userName = args[0];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listKeys(iam, userName);
        System.out.println("Done");
        iam.close();
    }

    public static void listKeys(IamClient iam, String userName) {
        try {
            boolean done = false;
            String newMarker = null;

            while (!done) {
                ListAccessKeysResponse response;

                if (newMarker == null) {
                    ListAccessKeysRequest request =
ListAccessKeysRequest.builder()
                            .userName(userName)
                            .build();

                    response = iam.listAccessKeys(request);

                } else {
                    ListAccessKeysRequest request =
ListAccessKeysRequest.builder()
                            .userName(userName)
                            .marker(newMarker)
                            .build();

                    response = iam.listAccessKeys(request);
                }

                for (AccessKeyMetadata metadata : response.accessKeyMetadata()) {
                    System.out.format("Retrieved access key %s",
metadata.accessKeyId());
                }

                if (!response.isTruncated()) {
```

```
                    done = true;
                } else {
                    newMarker = response.marker();
                }
            }

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca le chiavi di accesso.

```
import { ListAccessKeysCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 * @param {string} userName
 */
export async function* listAccessKeys(userName) {
  const command = new ListAccessKeysCommand({
```

```
    MaxItems: 5,
    UserName: userName,
  });

  /**
   * @type {import("@aws-sdk/client-iam").ListAccessKeysCommandOutput |
 undefined}
   */
  let response = await client.send(command);

  while (response?.AccessKeyMetadata?.length) {
    for (const key of response.AccessKeyMetadata) {
      yield key;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListAccessKeysCommand({
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
```

```
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  MaxItems: 5,
  UserName: "IAM_USER_NAME",
};

iam.listAccessKeys(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listKeys(userNameVal: String?) {
    val request =
        ListAccessKeysRequest {
            userName = userNameVal
        }
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAccessKeys(request)
        response.accessKeyMetadata?.forEach { md ->
            println("Retrieved access key ${md.accessKeyId}")
```

```
        }
    }
}
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando elenca le chiavi di accesso per l'utente IAM denominato**Bob**. Tieni presente che non puoi elencare le chiavi di accesso segrete per gli utenti IAM. Se le chiavi di accesso segrete vengono perse, è necessario creare nuove chiavi di accesso con il **`New-IAMAccessKey`** cmdlet.

```
Get-IAMAccessKey -UserName "Bob"
```

Output:

```
AccessKeyId               CreateDate                 Status
 UserName
-----------               ----------                 ------
 --------
AKIAIOSFODNN7EXAMPLE      12/3/2014 10:53:41 AM      Active             Bob
AKIAI44QH8DHBEXAMPLE      6/6/2013 8:42:26 PM        Inactive           Bob
```

- Per i dettagli sull'API, vedere ListAccessKeys in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_keys(user_name):
    """
    Lists the keys owned by the specified user.

    :param user_name: The name of the user.
    :return: The list of keys owned by the user.
    """
    try:
        keys = list(iam.User(user_name).access_keys.all())
        logger.info("Got %s access keys for %s.", len(keys), user_name)
    except ClientError:
        logger.exception("Couldn't get access keys for %s.", user_name)
        raise
    else:
        return keys
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, disattiva ed elimina le chiavi di accesso.

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end
```

```
  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity => e
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded => e
    @logger.error("Error creating access key: limit exceeded. Cannot create
more.")
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
```

```
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: "Inactive"
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta ListAccessKeys in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListAccountAliases`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAccountAliases`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestisci il tuo account

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool
AwsDoc::IAM::listAccountAliases(const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListAccountAliasesRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListAccountAliases(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list account aliases: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        const auto &aliases = outcome.GetResult().GetAccountAliases();
        if (!header) {
            if (aliases.size() == 0) {
                std::cout << "Account has no aliases" << std::endl;
                break;
            }
            std::cout << std::left << std::setw(32) << "Alias" << std::endl;
            header = true;
        }

        for (const auto &alias: aliases) {
            std::cout << std::left << std::setw(32) << alias << std::endl;
        }
```

```
        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for C++ API Reference.

CLI

AWS CLI

Elencare gli alias di un account

Il comando `list-account-aliases` seguente elenca gli alias per l'account corrente.

```
aws iam list-account-aliases
```

Output:

```
{
    "AccountAliases": [
    "mycompany"
    ]
}
```

Per ulteriori informazioni, consulta l'ID AWS dell'account e il relativo alias nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta ListAccountAliases in AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListAccountAliasesResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListAccountAliases {
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listAliases(iam);
        System.out.println("Done");
        iam.close();
    }

    public static void listAliases(IamClient iam) {
        try {
            ListAccountAliasesResponse response = iam.listAccountAliases();
            for (String alias : response.accountAliases()) {
                System.out.printf("Retrieved account alias %s", alias);
            }
```

```
        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca gli alias di un account.

```
import { ListAccountAliasesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 */
export async function* listAccountAliases() {
  const command = new ListAccountAliasesCommand({ MaxItems: 5 });

  let response = await client.send(command);

  while (response.AccountAliases?.length) {
    for (const alias of response.AccountAliases) {
      yield alias;
    }
```

```
    if (response.IsTruncated) {
      response = await client.send(
        new ListAccountAliasesCommand({
          Marker: response.Marker,
          MaxItems: 5,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.listAccountAliases({ MaxItems: 10 }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listAliases() {
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAccountAliases(ListAccountAliasesRequest {})
        response.accountAliases?.forEach { alias ->
            println("Retrieved account alias $alias")
        }
    }
}
```

- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce l'alias dell'account per. Account AWS

```
Get-IAMAccountAlias
```

Output:

```
ExampleCo
```

- Per i dettagli sull'API, vedere ListAccountAliases in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_aliases():
    """
    Gets the list of aliases for the current account. An account has at most one
 alias.

    :return: The list of aliases for the account.
    """
    try:
        response = iam.meta.client.list_account_aliases()
        aliases = response["AccountAliases"]
        if len(aliases) > 0:
            logger.info("Got aliases for your account: %s.", ",".join(aliases))
        else:
            logger.info("Got no aliases for your account.")
    except ClientError:
        logger.exception("Couldn't list aliases for your account.")
        raise
    else:
        return response["AccountAliases"]
```

- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, crea ed elimina gli alias degli account.

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info("Account aliases are:")
      response.account_aliases.each { |account_alias| @logger.info("
 #{account_alias}") }
    else
      @logger.info("No account aliases found.")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
```

```
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta ListAccountAliases in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListAttachedGroupPolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedGroupPolicies`.

CLI

AWS CLI

Per elencare tutte le politiche gestite allegate al gruppo specificato

Questo esempio restituisce i nomi e gli ARN delle politiche gestite allegate al gruppo IAM denominato `Admins` nell' AWS account.

```
aws iam list-attached-group-policies \
    --group-name Admins
```

Output:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedGroupPolicies AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al gruppo IAM denominato **`Admins`** nell' AWS account. Per visualizzare l'elenco delle politiche in linea incorporate nel gruppo, usa il **`Get-IAMGroupPolicyList`** comando.

```
Get-IAMAttachedGroupPolicyList -GroupName "Admins"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/SecurityAudit                     SecurityAudit
arn:aws:iam::aws:policy/AdministratorAccess               AdministratorAccess
```

- Per i dettagli sull'API, vedere ListAttachedGroupPoliciesin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListAttachedRolePolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedRolePolicies`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Come elencare tutte le policy gestite collegate al ruolo specificato

Questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al ruolo IAM denominato `SecurityAuditRole` nell' AWS account.

```
aws iam list-attached-role-policies \
    --role-name SecurityAuditRole
```

Output:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedRolePolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
```

```
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(roleName string)
 ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(context.TODO(),
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca le policy collegate a un ruolo.

```
import {
  ListAttachedRolePoliciesCommand,
  IAMClient,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 * @param {string} roleName
 */
export async function* listAttachedRolePolicies(roleName) {
  const command = new ListAttachedRolePoliciesCommand({
    RoleName: roleName,
  });

  let response = await client.send(command);

  while (response.AttachedPolicies?.length) {
    for (const policy of response.AttachedPolicies) {
      yield policy;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListAttachedRolePoliciesCommand({
          RoleName: roleName,
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listAttachedRolePolicies($roleName, $pathPrefix = "", $marker
 = "", $maxItems = 0)
    {
        $listAttachRolePoliciesArguments = ['RoleName' => $roleName];
        if ($pathPrefix) {
            $listAttachRolePoliciesArguments['PathPrefix'] = $pathPrefix;
        }
        if ($marker) {
            $listAttachRolePoliciesArguments['Marker'] = $marker;
        }
        if ($maxItems) {
            $listAttachRolePoliciesArguments['MaxItems'] = $maxItems;
        }
        return $this->iamClient-
>listAttachedRolePolicies($listAttachRolePoliciesArguments);
    }
```

- Per i dettagli sull'API, ListAttachedRolePoliciesconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite allegate al ruolo IAM denominato **`SecurityAuditRole`** nell' AWS account. Per visualizzare l'elenco delle politiche in linea incorporate nel ruolo, usa il **`Get-IAMRolePolicyList`** comando.

```
Get-IAMAttachedRolePolicyList -RoleName "SecurityAuditRole"
```

Output:

```
PolicyArn                                                 PolicyName
---------                                                 ----------
arn:aws:iam::aws:policy/SecurityAudit                     SecurityAudit
```

- Per i dettagli sull'API, vedere ListAttachedRolePoliciesin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_attached_policies(role_name):
    """
    Lists policies attached to a role.

    :param role_name: The name of the role to query.
    """
    try:
        role = iam.Role(role_name)
        for policy in role.attached_policies.all():
            logger.info("Got policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't list attached policies for %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta ListAttachedRolePolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, allega e scollega le politiche relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
```

```
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
```

```
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta AWS SDK for Ruby API ListAttachedRolePoliciesReference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_attached_role_policies(
    client: &iamClient,
    role_name: String,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListAttachedRolePoliciesOutput,
 SdkError<ListAttachedRolePoliciesError>> {
    let response = client
        .list_attached_role_policies()
        .role_name(role_name)
        .set_path_prefix(path_prefix)
        .set_marker(marker)
```

```
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListAttachedRolePoliciesall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// Returns a list of AWS Identity and Access Management (IAM) policies
   /// that are attached to the role.
   ///
   /// - Parameter role: The IAM role to return the policy list for.
   ///
   /// - Returns: An array of `IAMClientTypes.AttachedPolicy` objects
   ///   describing each managed policy that's attached to the role.
   public func listAttachedRolePolicies(role: String) async throws ->
[IAMClientTypes.AttachedPolicy] {
       var policyList: [IAMClientTypes.AttachedPolicy] = []
       var marker: String? = nil
       var isTruncated: Bool
```

```
        repeat {
            let input = ListAttachedRolePoliciesInput(
                marker: marker,
                roleName: role
            )
            let output = try await client.listAttachedRolePolicies(input: input)

            guard let attachedPolicies = output.attachedPolicies else {
                return policyList
            }

            for attachedPolicy in attachedPolicies {
                policyList.append(attachedPolicy)
            }
            marker = output.marker
            isTruncated = output.isTruncated
        } while isTruncated == true
        return policyList
    }
```

- Per i dettagli sull'API, consulta la ListAttachedRolePoliciesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListAttachedUserPolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListAttachedUserPolicies`.

CLI

AWS CLI

Per elencare tutte le politiche gestite allegate all'utente specificato

Questo comando restituisce i nomi e gli ARN delle politiche gestite per l'utente IAM indicato Bob nell' AWS account.

```
aws iam list-attached-user-policies \
    --user-name Bob
```

Output:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListAttachedUserPolicies AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i nomi e gli ARN delle politiche gestite per l'utente IAM indicato **Bob** nell' AWS account. Per visualizzare l'elenco delle politiche in linea incorporate nell'utente IAM, usa il **Get-IAMUserPolicyList** comando.

```
Get-IAMAttachedUserPolicyList -UserName "Bob"
```

Output:

```
PolicyArn                                                PolicyName
---------                                                ----------
arn:aws:iam::aws:policy/TesterPolicy                     TesterPolicy
```

- Per i dettagli sull'API, vedere ListAttachedUserPoliciesin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListEntitiesForPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListEntitiesForPolicy`.

CLI

AWS CLI

Per elencare tutti gli utenti, i gruppi e i ruoli a cui è allegata la politica gestita specificata

Questo esempio restituisce un elenco di gruppi, ruoli e utenti IAM a cui è `arn:aws:iam::123456789012:policy/TestPolicy` associata la policy.

```
aws iam list-entities-for-policy \
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Output:

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
```

```
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListEntitiesForPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce un elenco di gruppi, ruoli e utenti IAM a cui è **`arn:aws:iam::123456789012:policy/TestPolicy`** associata la policy.

```
Get-IAMEntitiesForPolicy -PolicyArn "arn:aws:iam::123456789012:policy/TestPolicy"
```

Output:

```
IsTruncated  : False
Marker       :
PolicyGroups : {}
PolicyRoles  : {testRole}
PolicyUsers  : {Bob, Theresa}
```

- Per i dettagli sull'API, vedere ListEntitiesForPolicyin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListGroupPolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroupPolicies`.

CLI

AWS CLI

Per elencare tutte le politiche in linea allegate al gruppo specificato

Il `list-group-policies` comando seguente elenca i nomi delle politiche in linea allegate al gruppo IAM denominato `Admins` nell'account corrente.

```
aws iam list-group-policies \
    --group-name Admins
```

Output:

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sulle API, consulta ListGroupPolicies in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce un elenco delle politiche in linea incorporate nel gruppo**`Testers`**. Per ottenere le politiche gestite allegate al gruppo, utilizzare il comando**`Get-IAMAttachedGroupPolicyList`**.

```
Get-IAMGroupPolicyList -GroupName Testers
```

Output:

```
Deny-Assume-S3-Role-In-Production
PowerUserAccess-Testers
```

- Per i dettagli sull'API, vedere ListGroupPolicies in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListGroups`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroups`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }
```

- Per i dettagli sull'API, consulta la ListGroupssezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Come elencare i gruppi IAM per l'account corrente

Il comando `list-groups` seguente elenca i gruppi IAM nell'account corrente.

```
aws iam list-groups
```

Output:

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-06-04T20:27:27.972Z",
            "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "GroupName": "Admins"
        },
        {
            "Path": "/",
            "CreateDate": "2013-04-16T20:30:42Z",
            "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
            "GroupName": "S3-Admins"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListGroups AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// GroupWrapper encapsulates AWS Identity and Access Management (IAM) group
 actions
// used in the examples.
// It contains an IAM service client that is used to perform group actions.
type GroupWrapper struct {
 IamClient *iam.Client
}



// ListGroups lists up to maxGroups number of groups.
func (wrapper GroupWrapper) ListGroups(maxGroups int32) ([]types.Group, error) {
 var groups []types.Group
 result, err := wrapper.IamClient.ListGroups(context.TODO(),
 &iam.ListGroupsInput{
  MaxItems: aws.Int32(maxGroups),
 })
 if err != nil {
  log.Printf("Couldn't list groups. Here's why: %v\n", err)
 } else {
  groups = result.Groups
 }
 return groups, err
}
```

- Per i dettagli sull'API, consulta la ListGroupssezione AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca i gruppi.

```
import { ListGroupsCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 */
export async function* listGroups() {
  const command = new ListGroupsCommand({
    MaxItems: 10,
  });

  let response = await client.send(command);

  while (response.Groups?.length) {
    for (const group of response.Groups) {
      yield group;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListGroupsCommand({
          Marker: response.Marker,
          MaxItems: 10,
        }),
      );
    } else {
      break;
```

```
    }
  }
}
```

- Per i dettagli sull'API, consulta la ListGroupssezione AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listGroups($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listGroupsArguments = [];
        if ($pathPrefix) {
            $listGroupsArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listGroupsArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listGroupsArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listGroups($listGroupsArguments);
    }
```

- Per i dettagli sull'API, consulta la ListGroupssezione AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce una raccolta di tutti i gruppi IAM definiti nella versione corrente Account AWS.

```
Get-IAMGroupList
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE2
GroupName  : Developers
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE3
GroupName  : Testers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_groups(count):
    """
    Lists the specified number of groups for the account.

    :param count: The number of groups to list.
    """
    try:
        for group in iam.groups.limit(count):
            logger.info("Group: %s", group.name)
    except ClientError:
        logger.exception("Couldn't list groups for the account.")
        raise
```

- Per i dettagli sull'API, consulta ListGroups AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# A class to manage IAM operations via the AWS SDK client
class IamGroupManager
  # Initializes the IamGroupManager class
  # @param iam_client [Aws::IAM::Client] An instance of the IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of groups for the account.
  # @param count [Integer] The maximum number of groups to list.
  # @return [Aws::IAM::Client::Response]
  def list_groups(count)
```

```
    response = @iam_client.list_groups(max_items: count)
    response.groups.each do |group|
      @logger.info("\t#{group.group_name}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list groups for the account. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- Per i dettagli sull'API, consulta la ListGroupssezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_groups(
    client: &iamClient,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListGroupsOutput, SdkError<ListGroupsError>> {
    let response = client
        .list_groups()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sulle API, consulta la ListGroupsguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func listGroups() async throws -> [String] {
    var groupList: [String] = []
    var marker: String? = nil
    var isTruncated: Bool

    repeat {
        let input = ListGroupsInput(marker: marker)
        let output = try await client.listGroups(input: input)

        guard let groups = output.groups else {
            return groupList
        }

        for group in groups {
            if let name = group.groupName {
                groupList.append(name)
            }
        }
        marker = output.marker
        isTruncated = output.isTruncated
```

```
        } while isTruncated == true
        return groupList
    }
```

- Per i dettagli sull'API, consulta la ListGroupsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListGroupsForUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListGroupsForUser`.

CLI

AWS CLI

Per elencare i gruppi a cui appartiene un utente IAM

Il `list-groups-for-user` comando seguente mostra i gruppi a cui Bob appartiene l'utente IAM denominato.

```
aws iam list-groups-for-user \
    --user-name Bob
```

Output:

```
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:18:08Z",
            "GroupId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/Admin",
            "GroupName": "Admin"
        },
        {
            "Path": "/",
            "CreateDate": "2013-05-06T01:37:28Z",
            "GroupId": "AKIAI44QH8DHBEXAMPLE",
```

```
            "Arn": "arn:aws:iam::123456789012:group/s3-Users",
            "GroupName": "s3-Users"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione di gruppi di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListGroupsForUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce l'elenco dei gruppi IAM a cui **David** appartiene l'utente IAM.

```
Get-IAMGroupForUser -UserName David
```

Output:

```
Arn        : arn:aws:iam::123456789012:group/Administrators
CreateDate : 10/20/2014 10:06:24 AM
GroupId    : 6WCH4TRY3KIHIEXAMPLE1
GroupName  : Administrators
Path       : /

Arn        : arn:aws:iam::123456789012:group/Testers
CreateDate : 12/10/2014 3:39:11 PM
GroupId    : RHNZZGQJ7QHMAEXAMPLE2
GroupName  : Testers
Path       : /

Arn        : arn:aws:iam::123456789012:group/Developers
CreateDate : 12/10/2014 3:38:55 PM
GroupId    : ZU2EOWMK6WBZOEXAMPLE3
GroupName  : Developers
Path       : /
```

- Per i dettagli sull'API, vedere ListGroupsForUserin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListInstanceProfiles`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListInstanceProfiles`.

CLI

AWS CLI

Per elencare i profili di istanza per l'account

Il `list-instance-profiles` comando seguente elenca i profili di istanza associati all'account corrente.

```
aws iam list-instance-profiles
```

Output:

```
{
    "InstanceProfiles": [
        {
            "Path": "/",
            "InstanceProfileName": "example-dev-role",
            "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
            "CreateDate": "2023-09-21T18:17:41+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-dev-role",
                    "RoleId": "AROAJ52OTH4H7LEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
                    "CreateDate": "2023-09-21T18:17:40+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
```

```
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
            ]
        },
        {
            "Path": "/",
            "InstanceProfileName": "example-s3-role",
            "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
            "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
            "CreateDate": "2023-09-21T18:18:50+00:00",
            "Roles": [
                {
                    "Path": "/",
                    "RoleName": "example-s3-role",
                    "RoleId": "AROAINUBC5O7XLEXAMPLE",
                    "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
                    "CreateDate": "2023-09-21T18:18:49+00:00",
                    "AssumeRolePolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                    "Service": "ec2.amazonaws.com"
                                },
                                "Action": "sts:AssumeRole"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListInstanceProfiles in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce una raccolta dei profili di istanza definiti nella versione corrente Account AWS.

```
Get-IAMInstanceProfileList
```

Output:

```
Arn                 : arn:aws:iam::123456789012:instance-profile/ec2instancerole
CreateDate          : 2/17/2015 2:49:04 PM
InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
InstanceProfileName : ec2instancerole
Path                : /
Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfiles in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListInstanceProfilesForRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListInstanceProfilesForRole`.

CLI

AWS CLI

Per elencare i profili di istanza per un ruolo IAM

Il `list-instance-profiles-for-role` comando seguente elenca i profili di istanza associati al ruolo`Test-Role`.

```
aws iam list-instance-profiles-for-role \
    --role-name Test-Role
```

Output:

```
{
    "InstanceProfiles": [
        {
            "InstanceProfileId": "AIDGPMS9RO4H3FEXAMPLE",
            "Roles": [
                {
                    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                    "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
                    "CreateDate": "2013-06-07T20:42:15Z",
                    "RoleName": "Test-Role",
                    "Path": "/",
                    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
                }
            ],
            "CreateDate": "2013-06-07T21:05:24Z",
            "InstanceProfileName": "ExampleInstanceProfile",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListInstanceProfilesForRole in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce i dettagli del profilo dell'istanza associato al ruolo**`ec2instancerole`**.

```
Get-IAMInstanceProfileForRole -RoleName ec2instancerole
```

Output:

```
      Arn                 : arn:aws:iam::123456789012:instance-profile/
ec2instancerole
      CreateDate          : 2/17/2015 2:49:04 PM
      InstanceProfileId   : HH36PTZQJUR32EXAMPLE1
      InstanceProfileName : ec2instancerole
      Path                : /
      Roles               : {ec2instancerole}
```

- Per i dettagli sull'API, vedere ListInstanceProfilesForRole in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListMfaDevices`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListMfaDevices`.

CLI

AWS CLI

Per elencare tutti i dispositivi MFA per un utente specificato

Questo esempio restituisce dettagli sul dispositivo MFA assegnato all'utente IAM. Bob

```
aws iam list-mfa-devices \
    --user-name Bob
```

Output:

```
{
    "MFADevices": [
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",
            "EnableDate": "2019-10-28T20:37:09+00:00"
        },
        {
            "UserName": "Bob",
```

```
            "SerialNumber": "GAKT12345678",
            "EnableDate": "2023-02-18T21:44:42+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",
            "EnableDate": "2023-09-19T02:25:35+00:00"
        },
        {
            "UserName": "Bob",
            "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",
            "EnableDate": "2023-09-19T01:49:18+00:00"
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta ListMfaDevices in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce i dettagli sul dispositivo MFA assegnato all'utente IAM. **`David`** In questo esempio si può dire che si tratta di un dispositivo virtuale perché **`SerialNumber`** è un ARN anziché il numero di serie effettivo di un dispositivo fisico.

```
Get-IAMMFADevice -UserName David
```

Output:

```
EnableDate                SerialNumber                          UserName
----------                ------------                          --------
4/8/2015 9:41:10 AM       arn:aws:iam::123456789012:mfa/David   David
```

- Per i dettagli sull'API, vedere ListMfaDevices in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListOpenIdConnectProviders`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListOpenIdConnectProviders`.

CLI

AWS CLI

Per elencare informazioni sui provider OpenID Connect presenti nell'account AWS

Questo esempio restituisce un elenco di ARNS di tutti i provider OpenID Connect definiti AWS nell'account corrente.

```
aws iam list-open-id-connect-providers
```

Output:

```
{
    "OpenIDConnectProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta ListOpenIdConnectProviders in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce un elenco di ARNS di tutti i provider OpenID Connect definiti nella Account AWS versione corrente.

```
Get-IAMOpenIDConnectProviderList
```

Output:

```
Arn
---
arn:aws:iam::123456789012:oidc-provider/server.example.com
arn:aws:iam::123456789012:oidc-provider/another.provider.com
```

- Per i dettagli sull'API, vedere ListOpenIdConnectProviders in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListPolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListPolicies`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione delle policy

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
```

```
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listPolicies(const Aws::Client::ClientConfiguration
 &clientConfig) {
    const Aws::String DATE_FORMAT("%Y-%m-%d");
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListPoliciesRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListPolicies(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list iam policies: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
```

```
        }

        if (!header) {
            std::cout << std::left << std::setw(55) << "Name" <<
                      std::setw(30) << "ID" << std::setw(80) << "Arn" <<
                      std::setw(64) << "Description" << std::setw(12) <<
                      "CreateDate" << std::endl;
            header = true;
        }

        const auto &policies = outcome.GetResult().GetPolicies();
        for (const auto &policy: policies) {
            std::cout << std::left << std::setw(55) <<
                      policy.GetPolicyName() << std::setw(30) <<
                      policy.GetPolicyId() << std::setw(80) << policy.GetArn() <<
                      std::setw(64) << policy.GetDescription() << std::setw(12)
 <<
                      policy.GetCreateDate().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Per elencare le politiche gestite disponibili per il tuo AWS account

Questo esempio restituisce una raccolta delle prime due politiche gestite disponibili nell' AWS account corrente.

```
aws iam list-policies \
    --max-items 3
```

Output:

```
{
    "Policies": [
        {
            "PolicyName": "AWSCloudTrailAccessPolicy",
            "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2019-09-04T17:43:42+00:00",
            "UpdateDate": "2019-09-04T17:43:42+00:00"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 6,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v5",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:47+00:00",
            "UpdateDate": "2023-07-06T22:04:00+00:00"
        }
```

```
    ],
    "NextToken": "EXAMPLErZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiA4fQ=="
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListPolicies AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(maxPolicies int32) ([]types.Policy,
 error) {
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(context.TODO(),
 &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
  policies = result.Policies
```

```
 }
 return policies, err
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca le policy.

```
import { ListPoliciesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listPolicies() {
  const command = new ListPoliciesCommand({
    MaxItems: 10,
    OnlyAttached: false,
    // List only the customer managed policies in your Amazon Web Services
 account.
    Scope: "Local",
  });

  let response = await client.send(command);
```

```
  while (response.Policies?.length) {
    for (const policy of response.Policies) {
      yield policy;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListPoliciesCommand({
          Marker: response.Marker,
          MaxItems: 10,
          OnlyAttached: false,
          Scope: "Local",
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listPolicies($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listPoliciesArguments = [];
        if ($pathPrefix) {
            $listPoliciesArguments["PathPrefix"] = $pathPrefix;
        }
```

```
        if ($marker) {
            $listPoliciesArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listPoliciesArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listPolicies($listPoliciesArguments);
    }
```

- Per i dettagli sull'API, ListPoliciesconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio restituisce una raccolta delle prime tre politiche gestite disponibili nell' AWS account corrente. Poiché non **`-scope`** è specificato, per **`all`** impostazione predefinita include sia le politiche AWS gestite che quelle gestite dai clienti.

```
Get-IAMPolicyList -MaxItem 3
```

Output:

```
Arn              : arn:aws:iam::aws:policy/AWSDirectConnectReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:08 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : Z27SI6FQMGNQ2EXAMPLE1
PolicyName       : AWSDirectConnectReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:08 AM

Arn              : arn:aws:iam::aws:policy/AmazonGlacierReadOnlyAccess
AttachmentCount  : 0
CreateDate       : 2/6/2015 10:40:27 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
```

```
Path             : /
PolicyId         : NJKMU274MET4EEXAMPLE2
PolicyName       : AmazonGlacierReadOnlyAccess
UpdateDate       : 2/6/2015 10:40:27 AM

Arn              : arn:aws:iam::aws:policy/AWSMarketplaceFullAccess
AttachmentCount  : 0
CreateDate       : 2/11/2015 9:21:45 AM
DefaultVersionId : v1
Description      :
IsAttachable     : True
Path             : /
PolicyId         : 5ULJSO2FYVPYGEXAMPLE3
PolicyName       : AWSMarketplaceFullAccess
UpdateDate       : 2/11/2015 9:21:45 AM
```

Esempio 2: Questo esempio restituisce una raccolta delle prime due politiche gestite dai clienti disponibili nell'account corrente AWS . Viene utilizzato **`-Scope local`** per limitare l'output alle sole politiche gestite dal cliente.

```
Get-IAMPolicyList -Scope local -MaxItem 2
```

Output:

```
Arn              : arn:aws:iam::123456789012:policy/MyLocalPolicy
AttachmentCount  : 0
CreateDate       : 2/12/2015 9:39:09 AM
DefaultVersionId : v2
Description      :
IsAttachable     : True
Path             : /
PolicyId         : SQVCBLC4VAOUCEXAMPLE4
PolicyName       : MyLocalPolicy
UpdateDate       : 2/12/2015 9:39:53 AM

Arn              : arn:aws:iam::123456789012:policy/policyforec2instancerole
AttachmentCount  : 1
CreateDate       : 2/17/2015 2:51:38 PM
DefaultVersionId : v11
Description      :
IsAttachable     : True
Path             : /
```

```
PolicyId         : X5JPBLJH2Z2SOEXAMPLE5
PolicyName       : policyforec2instancerole
UpdateDate       : 2/18/2015 8:52:31 AM
```

- Per i dettagli sull'API, vedere ListPoliciesin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_policies(scope):
    """
    Lists the policies in the current account.

    :param scope: Limits the kinds of policies that are returned. For example,
                  'Local' specifies that only locally managed policies are
 returned.
    :return: The list of policies.
    """
    try:
        policies = list(iam.policies.filter(Scope=scope))
        logger.info("Got %s policies in scope '%s'.", len(policies), scope)
    except ClientError:
        logger.exception("Couldn't get policies for scope '%s'.", scope)
        raise
    else:
        return policies
```

- Per i dettagli sull'API, consulta ListPolicies AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo modulo di esempio elenca, crea, allega e scollega le politiche relative ai ruoli.

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
```

```
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not
exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
```

```
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
    false
  end
end
```

- Per i dettagli sulle API, consulta la sezione AWS SDK for Ruby API ListPoliciesReference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_policies(
    client: iamClient,
    path_prefix: String,
) -> Result<Vec<String>, SdkError<ListPoliciesError>> {
    let list_policies = client
        .list_policies()
        .path_prefix(path_prefix)
        .scope(PolicyScopeType::Local)
        .into_paginator()
        .items()
        .send()
        .try_collect()
        .await?;
```

```
    let policy_names = list_policies
        .into_iter()
        .map(|p| {
            let name = p
                .policy_name
                .unwrap_or_else(|| "Missing Policy Name".to_string());
            println!("{}", name);
            name
        })
        .collect();

    Ok(policy_names)
}
```

- Per i dettagli sulle API, consulta il riferimento ListPoliciesall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func listPolicies() async throws -> [MyPolicyRecord] {
        var policyList: [MyPolicyRecord] = []
        var marker: String? = nil
        var isTruncated: Bool

        repeat {
```

```
            let input = ListPoliciesInput(marker: marker)
            let output = try await client.listPolicies(input: input)

            guard let policies = output.policies else {
                return policyList
            }

            for policy in policies {
                guard   let name = policy.policyName,
                        let id = policy.policyId,
                        let arn = policy.arn else {
                    throw ServiceHandlerError.noSuchPolicy
                }
                policyList.append(MyPolicyRecord(name: name, id: id, arn: arn))
            }
            marker = output.marker
            isTruncated = output.isTruncated
        } while isTruncated == true
        return policyList
    }
```

- Per i dettagli sull'API, consulta la ListPoliciesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListPolicyVersions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListPolicyVersions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Per elencare informazioni sulle versioni della politica gestita specificata

Questo esempio restituisce l'elenco delle versioni disponibili della politica il cui ARN è. `arn:aws:iam::123456789012:policy/MySamplePolicy`

```
aws iam list-policy-versions \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{
    "IsTruncated": false,
    "Versions": [
        {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2015-06-02T23:19:44Z"
        },
        {
        "VersionId": "v1",
        "IsDefaultVersion": false,
        "CreateDate": "2015-06-02T22:30:47Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta ListPolicyVersions in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce l'elenco delle versioni disponibili della politica il cui ARN è. **`arn:aws:iam::123456789012:policy/MyManagedPolicy`** Per ottenere il documento

relativo alla policy per una versione specifica, utilizzate il **`Get-IAMPolicyVersion`** comando e specificate quella desiderata. **`VersionId`**

```
Get-IAMPolicyVersionList -PolicyArn arn:aws:iam::123456789012:policy/
MyManagedPolicy
```

Output:

```
CreateDate                  Document                IsDefaultVersion
       VersionId
----------                  --------                ---------------
       ---------
2/12/2015 9:39:53 AM                                True
       v2
2/12/2015 9:39:09 AM                                False
       v1
```

- Per i dettagli sull'API, vedere ListPolicyVersions in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListRolePolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRolePolicies`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
```

```
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
 _IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
 roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come elencare le policy colegate a un ruolo IAM

Il comando `list-role-policies` seguente elenca i nomi delle policy di autorizzazione per il ruolo IAM specificato.

```
aws iam list-role-policies \
    --role-name Test-Role
```

Output:

```
{
    "PolicyNames": [
        "ExamplePolicy"
    ]
```

```
}
```

Per consultare la policy di attendibilità collegata a un ruolo, usa il comando `get-role`. Per visualizzare i dettagli di una policy di autorizzazioni, usa il comando `get-role-policy`.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListRolePolicies in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(roleName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(context.TODO(),
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
```

```
 return policies, err
}
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca le policy.

```
import { ListRolePoliciesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 * @param {string} roleName
 */
export async function* listRolePolicies(roleName) {
  const command = new ListRolePoliciesCommand({
    RoleName: roleName,
    MaxItems: 10,
  });

  let response = await client.send(command);

  while (response.PolicyNames?.length) {
    for (const policyName of response.PolicyNames) {
      yield policyName;
```

```
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListRolePoliciesCommand({
          RoleName: roleName,
          MaxItems: 10,
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listRolePolicies($roleName, $marker = "", $maxItems = 0)
    {
        $listRolePoliciesArguments = ['RoleName' => $roleName];
        if ($marker) {
            $listRolePoliciesArguments['Marker'] = $marker;
        }
        if ($maxItems) {
            $listRolePoliciesArguments['MaxItems'] = $maxItems;
        }
        return $this->customWaiter(function () use ($listRolePoliciesArguments) {
```

```
            return $this->iamClient-
>listRolePolicies($listRolePoliciesArguments);
        });
    }
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio restituisce l'elenco dei nomi delle politiche in linea incorporate nel ruolo **`lamda_exec_role`** IAM. Per visualizzare i dettagli di una politica in linea, usa il comando. **`Get-IAMRolePolicy`**

```
Get-IAMRolePolicyList -RoleName lambda_exec_role
```

Output:

```
oneClick_lambda_exec_role_policy
```

- Per i dettagli sull'API, vedere ListRolePolicies in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_policies(role_name):
    """
    Lists inline policies for a role.
```

```
    :param role_name: The name of the role to query.
    """
    try:
        role = iam.Role(role_name)
        for policy in role.policies.all():
            logger.info("Got inline policy %s.", policy.name)
    except ClientError:
        logger.exception("Couldn't list inline policies for %s.", role_name)
        raise
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Lists policy ARNs attached to a role
#
# @param role_name [String] The name of the role
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_role_policies(
    client: &iamClient,
    role_name: &str,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListRolePoliciesOutput, SdkError<ListRolePoliciesError>> {
    let response = client
        .list_role_policies()
        .role_name(role_name)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;

    Ok(response)
}
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for Rust API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func listRolePolicies(role: String) async throws -> [String] {
    var policyList: [String] = []
    var marker: String? = nil
    var isTruncated: Bool

    repeat {
        let input = ListRolePoliciesInput(
            marker: marker,
            roleName: role
        )
        let output = try await client.listRolePolicies(input: input)

        guard let policies = output.policyNames else {
            return policyList
        }

        for policy in policies {
            policyList.append(policy)
        }
        marker = output.marker
        isTruncated = output.isTruncated
    } while isTruncated == true
    return policyList
}
```

- Per i dettagli sull'API, consulta ListRolePolicies in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListRoleTags`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRoleTags`.

CLI

AWS CLI

Per elencare i tag associati a un ruolo

Il `list-role-tags` comando seguente recupera l'elenco dei tag associati al ruolo specificato.

```
aws iam list-role-tags \
    --role-name production-role
```

Output:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta ListRoleTags in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio recupera il tag associato al ruolo.

```
Get-IAMRoleTagList -RoleName MyRoleName
```

- Per i dettagli sull'API, vedere ListRoleTags in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListRoles`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRoles`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Come elencare i ruoli IAM per l'account corrente

Il comando `list-roles` seguente elenca i ruoli IAM per l'account corrente.

```
aws iam list-roles
```

Output:

```
{
    "Roles": [
        {
            "Path": "/",
            "RoleName": "ExampleRole",
            "RoleId": "AROAJ52OTH4H7LEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
            "CreateDate": "2017-09-12T19:23:36+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/example_path/",
            "RoleName": "ExampleRoleWithPath",
            "RoleId": "AROAI4QRP7UFT7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
            "CreateDate": "2023-09-21T20:29:38+00:00",
            "AssumeRolePolicyDocument": {
```

```
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListRoles AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
```

```
func (wrapper RoleWrapper) ListRoles(maxRoles int32) ([]types.Role, error) {
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(context.TODO(),
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
 return roles, err
}
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca i ruoli.

```
import { ListRolesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listRoles() {
  const command = new ListRolesCommand({
```

```
    MaxItems: 10,
  });

  /**
   * @type {import("@aws-sdk/client-iam").ListRolesCommandOutput | undefined}
   */
  let response = await client.send(command);

  while (response?.Roles?.length) {
    for (const role of response.Roles) {
      yield role;
    }

    if (response.IsTruncated) {
      response = await client.send(
        new ListRolesCommand({
          Marker: response.Marker,
        }),
      );
    } else {
      break;
    }
  }
}
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    /**
```

```
     * @param string $pathPrefix
     * @param string $marker
     * @param int $maxItems
     * @return Result
     * $roles = $service->listRoles();
     */
    public function listRoles($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listRolesArguments = [];
        if ($pathPrefix) {
            $listRolesArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listRolesArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listRolesArguments["MaxItems"] = $maxItems;
        }
        return $this->iamClient->listRoles($listRolesArguments);
    }
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera un elenco di tutti i ruoli IAM in. Account AWS

```
Get-IAMRoleList
```

Esempio 2: Questo frammento di codice di esempio recupera un elenco di ruoli IAM nell' AWS account e li visualizza tre alla volta, e attende che tu premi Invio tra ogni gruppo. Passa il **`Marker`** valore della chiamata precedente per specificare dove deve iniziare il gruppo successivo.

```
$nextMarker = $null
Do
{
    $results = Get-IAMRoleList -MaxItem 3 -Marker $nextMarker
```

```
    $nextMarker = $AWSHistory.LastServiceResponse.Marker
    $results
    Read-Host
} while ($nextMarker -ne $null)
```

- Per i dettagli sull'API, vedere ListRolesin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_roles(count):
    """
    Lists the specified number of roles for the account.

    :param count: The number of roles to list.
    """
    try:
        roles = list(iam.roles.limit(count=count))
        for role in roles:
            logger.info("Role: %s", role.name)
    except ClientError:
        logger.exception("Couldn't list roles for the account.")
        raise
    else:
        return roles
```

- Per i dettagli sull'API, consulta ListRoles AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Lists IAM roles up to a specified count.
# @param count [Integer] the maximum number of roles to list.
# @return [Array<String>] the names of the roles.
def list_roles(count)
  role_names = []
  roles_counted = 0

  @iam_client.list_roles.each_page do |page|
    page.roles.each do |role|
      break if roles_counted >= count
      @logger.info("\t#{roles_counted + 1}: #{role.role_name}")
      role_names << role.role_name
      roles_counted += 1
    end
    break if roles_counted >= count
  end

  role_names
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't list roles for the account. Here's why:")
  @logger.error("\t#{e.code}: #{e.message}")
  raise
end
```

- Per i dettagli sull'API, ListRolesconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_roles(
    client: &iamClient,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListRolesOutput, SdkError<ListRolesError>> {
    let response = client
        .list_roles()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
        .await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta il riferimento ListRolesall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func listRoles() async throws -> [String] {
    var roleList: [String] = []
    var marker: String? = nil
    var isTruncated: Bool

    repeat {
        let input = ListRolesInput(marker: marker)
        let output = try await client.listRoles(input: input)

        guard let roles = output.roles else {
            return roleList
        }

        for role in roles {
            if let name = role.roleName {
                roleList.append(name)
            }
        }
        marker = output.marker
        isTruncated = output.isTruncated
    } while isTruncated == true
    return roleList
}
```

- Per i dettagli sull'API, consulta la ListRolesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListSAMLProviders`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListSAMLProviders`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API AWS SDK for .NET .

CLI

AWS CLI

Per elencare i provider SAML presenti nell'account AWS

Questo esempio recupera l'elenco dei provider SAML 2.0 creati nell'account corrente. AWS

```
aws iam list-saml-providers
```

Output:

```
{
    "SAMLProviderList": [
```

```
        {
            "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
            "ValidUntil": "2015-06-05T22:45:14Z",
            "CreateDate": "2015-06-05T22:45:14Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per informazioni dettagliate sull'API, consulta ListSAMLProviders nella Documentazione di riferimento dei comandi della AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}



// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders() ([]types.SAMLProviderListEntry,
 error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(context.TODO(),
 &iam.ListSAMLProvidersInput{})
```

```
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API AWS SDK for Go .

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca gli IdP SAML.

```
import { ListSAMLProvidersCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listSamlProviders = async () => {
  const command = new ListSAMLProvidersCommand({});

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API AWS SDK for JavaScript .

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listSAMLProviders()
    {
        return $this->iamClient->listSAMLProviders();
    }
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API AWS SDK for PHP .

PowerShell

Utensili per PowerShell

Esempio 1: questo esempio recupera l'elenco dei provider SAML 2.0 creati nella versione corrente. Account AWS Restituisce l'ARN, la data di creazione e la data di scadenza per ogni provider SAML.

```
Get-IAMSAMLProviderList
```

Output:

```
Arn                                                 CreateDate
   ValidUntil
---                                                 ----------
   ----------
arn:aws:iam::123456789012:saml-provider/SAMLADFS    12/23/2014 12:16:55 PM
   12/23/2114 12:16:54 PM
```

- Per i dettagli sull'API, vedere ListSAMLProviders in Cmdlet Reference.AWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_saml_providers(count):
    """
    Lists the SAML providers for the account.

    :param count: The maximum number of providers to list.
    """
    try:
        found = 0
        for provider in iam.saml_providers.limit(count):
            logger.info("Got SAML provider %s.", provider.arn)
            found += 1
        if found == 0:
            logger.info("Your account has no SAML providers.")
    except ClientError:
        logger.exception("Couldn't list SAML providers.")
        raise
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class SamlProviderLister
  # Initializes the SamlProviderLister with IAM client and a logger.
  # @param iam_client [Aws::IAM::Client] The IAM client object.
  # @param logger [Logger] The logger object for logging output.
  def initialize(iam_client, logger = Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of SAML providers for the account.
  # @param count [Integer] The maximum number of providers to list.
  # @return [Aws::IAM::Client::Response]
  def list_saml_providers(count)
    response = @iam_client.list_saml_providers
    response.saml_provider_list.take(count).each do |provider|
      @logger.info("\t#{provider.arn}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list SAML providers. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API AWS SDK for Ruby .

Rust

SDK per Rust

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_saml_providers(
    client: &Client,
) -> Result<ListSamlProvidersOutput, SdkError<ListSAMLProvidersError>> {
    let response = client.list_saml_providers().send().await?;

    Ok(response)
}
```

- Per informazioni dettagliate sull'API, consulta la sezione ListSAMLProviders nella Documentazione di riferimento dell'API SDK AWS per Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListServerCertificates`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListServerCertificates`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listServerCertificates(
        const Aws::Client::ClientConfiguration &clientConfig) {
    const Aws::String DATE_FORMAT = "%Y-%m-%d";

    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListServerCertificatesRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListServerCertificates(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list server certificates: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(55) << "Name" <<
                      std::setw(30) << "ID" << std::setw(80) << "Arn" <<
                      std::setw(14) << "UploadDate" << std::setw(14) <<
                      "ExpirationDate" << std::endl;
            header = true;
        }

        const auto &certificates =
                outcome.GetResult().GetServerCertificateMetadataList();

        for (const auto &certificate: certificates) {
            std::cout << std::left << std::setw(55) <<
                      certificate.GetServerCertificateName() << std::setw(30) <<
                      certificate.GetServerCertificateId() << std::setw(80) <<
                      certificate.GetArn() << std::setw(14) <<

 certificate.GetUploadDate().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::setw(14) <<

 certificate.GetExpiration().ToGmtString(DATE_FORMAT.c_str()) <<
                      std::endl;
        }

        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
```

```
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta ListServerCertificates in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per elencare i certificati del server presenti nel tuo AWS account

Il `list-server-certificates` comando seguente elenca tutti i certificati server archiviati e disponibili per l'uso nell' AWS account.

```
aws iam list-server-certificates
```

Output:

```
{
    "ServerCertificateMetadataList": [
        {
            "Path": "/",
            "ServerCertificateName": "myUpdatedServerCertificate",
            "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/
myUpdatedServerCertificate",
            "UploadDate": "2019-04-22T21:13:44+00:00",
            "Expiration": "2019-10-15T22:23:16+00:00"
        },
        {
            "Path": "/cloudfront/",
            "ServerCertificateName": "MyTestCert",
            "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyTestCert",
```

```
            "UploadDate": "2015-04-21T18:14:16+00:00",
            "Expiration": "2018-01-14T17:52:36+00:00"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListServerCertificates in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca i certificati.

```
import { ListServerCertificatesCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 * A generator function that handles paginated results.
 * The AWS SDK for JavaScript (v3) provides {@link https://docs.aws.amazon.com/
AWSJavaScriptSDK/v3/latest/index.html#paginators | paginator} functions to
 simplify this.
 *
 */
export async function* listServerCertificates() {
  const command = new ListServerCertificatesCommand({});
  let response = await client.send(command);

  while (response.ServerCertificateMetadataList?.length) {
    for await (const cert of response.ServerCertificateMetadataList) {
      yield cert;
    }
```

```
    if (response.IsTruncated) {
      response = await client.send(new ListServerCertificatesCommand({}));
    } else {
      break;
    }
  }
}
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta ListServerCertificates in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

iam.listServerCertificates({}, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.

- Per i dettagli sull'API, consulta ListServerCertificates in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera l'elenco dei certificati server che sono stati caricati nella versione corrente Account AWS.

```
Get-IAMServerCertificateList
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/Org1/Org2/
MyServerCertificate
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /Org1/Org2/
ServerCertificateId   : ASCAJIFEXAMPLE17HQZYW
ServerCertificateName : MyServerCertificate
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere ListServerCertificates in AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati del server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
```

```
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key,
                                          })
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info("No server certificates found.")
      return
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
```

```
    @logger.info("Server certificate name updated from '#{current_name}' to
 '#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta ListServerCertificates in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListSigningCertificates`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListSigningCertificates`.

CLI

AWS CLI

Per elencare i certificati di firma per un utente IAM

Il `list-signing-certificates` comando seguente elenca i certificati di firma per l'utente IAM denominatoBob.

```
aws iam list-signing-certificates \
    --user-name Bob
```

Output:

```
{
    "Certificates": [
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-
body>-----END CERTIFICATE-----",
            "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
            "UploadDate": "2013-06-06T21:40:08Z"
        }
    ]
}
```

Per ulteriori informazioni, consulta Gestire i certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta ListSigningCertificates in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera i dettagli sul certificato di firma associato all'utente denominato**Bob**.

```
Get-IAMSigningCertificate -UserName Bob
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
```

```
 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                  rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE

 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU
Status          : Active
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere ListSigningCertificates in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListUserPolicies`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserPolicies`.

CLI

AWS CLI

Per elencare le policy per un utente IAM

Il comando `list-user-policies` seguente elenca le policy collegate all'utente IAM denominato Bob.

```
aws iam list-user-policies \
    --user-name Bob
```

Output:

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
}
```

Per ulteriori informazioni, consulta Creating an IAM user in your AWS account nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta ListUserPolicies in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(userName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(context.TODO(),
 &iam.ListUserPoliciesInput{
```

```
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}
```

- Per i dettagli sull'API, consulta ListUserPolicies in AWS SDK for Go API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera l'elenco dei nomi delle politiche in linea incorporate nell'utente IAM denominato. **`David`**

```
Get-IAMUserPolicyList -UserName David
```

Output:

```
Davids_IAM_Admin_Policy
```

- Per i dettagli sull'API, vedere ListUserPolicies in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListUserTags`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserTags`.

CLI

AWS CLI

Per elencare i tag associati a un utente

Il `list-user-tags` comando seguente recupera l'elenco dei tag associati all'utente IAM specificato.

```
aws iam list-user-tags \
    --user-name alice
```

Output:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta ListUserTags in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio recupera il tag associato all'utente.

```
Get-IAMUserTagList -UserName joe
```

- Per i dettagli sull'API, vedere ListUserTags in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListUsers`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUsers`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#       The list of users names
#    And:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
##############################################################################
function iam_list_users() {
  local option OPTARG # Required to use getopts command in a function.
  local error_code
  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_list_users"
    echo "Lists the AWS Identity and Access Management (IAM) user in the
 account."
```

```
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "h" option; do
    case "${option}" in
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local response

  response=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
  error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-users operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta ListUsers AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::listUsers(const Aws::Client::ClientConfiguration &clientConfig)
 {
    const Aws::String DATE_FORMAT = "%Y-%m-%d";
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::ListUsersRequest request;

    bool done = false;
    bool header = false;
    while (!done) {
        auto outcome = iam.ListUsers(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed to list iam users:" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (!header) {
            std::cout << std::left << std::setw(32) << "Name" <<
                      std::setw(30) << "ID" << std::setw(64) << "Arn" <<
                      std::setw(20) << "CreateDate" << std::endl;
            header = true;
        }

        const auto &users = outcome.GetResult().GetUsers();
        for (const auto &user: users) {
            std::cout << std::left << std::setw(32) << user.GetUserName() <<
                      std::setw(30) << user.GetUserId() << std::setw(64) <<
                      user.GetArn() << std::setw(20) <<
                      user.GetCreateDate().ToGmtString(DATE_FORMAT.c_str())
                      << std::endl;
        }
```

```
        if (outcome.GetResult().GetIsTruncated()) {
            request.SetMarker(outcome.GetResult().GetMarker());
        }
        else {
            done = true;
        }
    }

    return true;
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per elencare gli utenti IAM

Il comando `list-users` seguente elenca gli utenti IAM nell'account corrente.

```
aws iam list-users
```

Output:

```
{
    "Users": [
        {
            "UserName": "Adele",
            "Path": "/",
            "CreateDate": "2013-03-07T05:14:48Z",
            "UserId": "AKIAI44QH8DHBEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Adele"
        },
        {
            "UserName": "Bob",
            "Path": "/",
            "CreateDate": "2012-09-21T23:03:13Z",
            "UserId": "AKIAIOSFODNN7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/Bob"
        }
    ]
```

```
}
```

Per ulteriori informazioni, consulta Elencazione degli utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta ListUsers AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(maxUsers int32) ([]types.User, error) {
 var users []types.User
 result, err := wrapper.IamClient.ListUsers(context.TODO(), &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
 return users, err
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.AttachedPermissionsBoundary;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.ListUsersRequest;
import software.amazon.awssdk.services.iam.model.ListUsersResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.User;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListUsers {
    public static void main(String[] args) {
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        listAllUsers(iam);
        System.out.println("Done");
        iam.close();
    }

    public static void listAllUsers(IamClient iam) {
```

```
        try {
            boolean done = false;
            String newMarker = null;
            while (!done) {
                ListUsersResponse response;
                if (newMarker == null) {
                    ListUsersRequest request =
 ListUsersRequest.builder().build();
                    response = iam.listUsers(request);
                } else {
                    ListUsersRequest request = ListUsersRequest.builder()
                            .marker(newMarker)
                            .build();

                    response = iam.listUsers(request);
                }

                for (User user : response.users()) {
                    System.out.format("\n Retrieved user %s", user.userName());
                    AttachedPermissionsBoundary permissionsBoundary =
 user.permissionsBoundary();
                    if (permissionsBoundary != null)
                        System.out.format("\n Permissions boundary details %s",

 permissionsBoundary.permissionsBoundaryTypeAsString());
                }

                if (!response.isTruncated()) {
                    done = true;
                } else {
                    newMarker = response.marker();
                }
            }

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca gli utenti.

```
import { ListUsersCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

export const listUsers = async () => {
  const command = new ListUsersCommand({ MaxItems: 10 });

  const response = await client.send(command);
  response.Users?.forEach(({ UserName, CreateDate }) => {
    console.log(`${UserName} created on: ${CreateDate}`);
  });
  return response;
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
```

```
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  MaxItems: 10,
};

iam.listUsers(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    var users = data.Users || [];
    users.forEach(function (user) {
      console.log("User " + user.UserName + " created", user.CreateDate);
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listAllUsers() {
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listUsers(ListUsersRequest { })
        response.users?.forEach { user ->
            println("Retrieved user ${user.userName}")
            val permissionsBoundary = user.permissionsBoundary
            if (permissionsBoundary != null) {
```

```
                println("Permissions boundary details
 ${permissionsBoundary.permissionsBoundaryType}")
            }
        }
    }
}
```

- Per i dettagli sull'API, ListUsersconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
$uuid = uniqid();
$service = new IAMService();

    public function listUsers($pathPrefix = "", $marker = "", $maxItems = 0)
    {
        $listUsersArguments = [];
        if ($pathPrefix) {
            $listUsersArguments["PathPrefix"] = $pathPrefix;
        }
        if ($marker) {
            $listUsersArguments["Marker"] = $marker;
        }
        if ($maxItems) {
            $listUsersArguments["MaxItems"] = $maxItems;
        }

        return $this->iamClient->listUsers($listUsersArguments);
    }
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera una raccolta di utenti nella cartella corrente Account AWS.

```
Get-IAMUserList
```

Output:

```
Arn              : arn:aws:iam::123456789012:user/Administrator
CreateDate       : 10/16/2014 9:03:09 AM
PasswordLastUsed : 3/4/2015 12:12:33 PM
Path             : /
UserId           : 7K3GJEANSKZF2EXAMPLE1
UserName         : Administrator

Arn              : arn:aws:iam::123456789012:user/Bob
CreateDate       : 4/6/2015 12:54:42 PM
PasswordLastUsed : 1/1/0001 12:00:00 AM
Path             : /
UserId           : L3EWNONDOM3YUEXAMPLE2
UserName         : bab

Arn              : arn:aws:iam::123456789012:user/David
CreateDate       : 12/10/2014 3:39:27 PM
PasswordLastUsed : 3/19/2015 8:44:04 AM
Path             : /
UserId           : Y4FKWQCXTA52QEXAMPLE3
UserName         : David
```

- Per i dettagli sull'API, vedere ListUsersin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def list_users():
    """
    Lists the users in the current account.

    :return: The list of users.
    """
    try:
        users = list(iam.users.all())
        logger.info("Got %s users.", len(users))
    except ClientError:
        logger.exception("Couldn't get users.")
        raise
    else:
        return users
```

- Per i dettagli sull'API, consulta ListUsers AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  # Lists all users in the AWS account
  #
  # @return [Array<Aws::IAM::Types::User>] An array of user objects
  def list_users
    users = []
    @iam_client.list_users.each_page do |page|
      page.users.each do |user|
        users << user
      end
    end
    users
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing users: #{e.message}")
    []
  end
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
pub async fn list_users(
    client: &iamClient,
    path_prefix: Option<String>,
    marker: Option<String>,
    max_items: Option<i32>,
) -> Result<ListUsersOutput, SdkError<ListUsersError>> {
    let response = client
        .list_users()
        .set_path_prefix(path_prefix)
        .set_marker(marker)
        .set_max_items(max_items)
        .send()
```

```
        .await?;
    Ok(response)
}
```

- Per i dettagli sulle API, consulta la ListUsersguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func listUsers() async throws -> [MyUserRecord] {
        var userList: [MyUserRecord] = []
        var marker: String? = nil
        var isTruncated: Bool

        repeat {
            let input = ListUsersInput(marker: marker)
            let output = try await client.listUsers(input: input)

            guard let users = output.users else {
                return userList
            }

            for user in users {
                if let id = user.userId, let name = user.userName {
                    userList.append(MyUserRecord(id: id, name: name))
                }
```

```
            }
            marker = output.marker
            isTruncated = output.isTruncated
        } while isTruncated == true
        return userList
    }
```

- Per i dettagli sull'API, consulta la ListUsersguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListVirtualMfaDevices`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListVirtualMfaDevices`.

CLI

AWS CLI

Per elencare i dispositivi MFA virtuali

Il `list-virtual-mfa-devices` comando seguente elenca i dispositivi MFA virtuali che sono stati configurati per l'account corrente.

```
aws iam list-virtual-mfa-devices
```

Output:

```
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
        },
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
        }
    ]
}
```

Per ulteriori informazioni, consulta Abilitazione di un dispositivo di autenticazione a più fattori (MFA) virtuale nella Guida per AWS l'utente IAM.

- Per i dettagli sull'API, consulta AWS CLI Command ListVirtualMfaDevicesReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio recupera una raccolta di dispositivi MFA virtuali assegnati agli utenti dell'account. AWS La **`User`** proprietà di ciascuno è un oggetto con i dettagli dell'utente IAM a cui è assegnato il dispositivo.

```
Get-IAMVirtualMFADevice -AssignmentStatus Assigned
```

Output:

```
Base32StringSeed :
EnableDate       : 4/13/2015 12:03:42 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/David
User             : Amazon.IdentityManagement.Model.User

Base32StringSeed :
EnableDate       : 4/13/2015 12:06:41 PM
QRCodePNG        :
SerialNumber     : arn:aws:iam::123456789012:mfa/root-account-mfa-device
User             : Amazon.IdentityManagement.Model.User
```

- Per i dettagli sull'API, vedere ListVirtualMfaDevicesin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutGroupPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutGroupPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Add or update an inline policy document that is embedded in an IAM group.
   /// </summary>
   /// <param name="groupName">The name of the IAM group.</param>
   /// <param name="policyName">The name of the IAM policy.</param>
   /// <param name="policyDocument">The policy document defining the IAM
policy.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> PutGroupPolicyAsync(string groupName, string
policyName, string policyDocument)
   {
       var request = new PutGroupPolicyRequest
       {
           GroupName = groupName,
           PolicyName = policyName,
           PolicyDocument = policyDocument
       };

       var response = await _IAMService.PutGroupPolicyAsync(request);
       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }
```

- Per i dettagli sull'API, consulta la PutGroupPolicy in AWS SDK for .NET API Reference.

CLI

AWS CLI

Come aggiungere una policy a un gruppo

Il comando `put-group-policy` seguente aggiunge una policy al gruppo IAM denominato `Admins`.

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

Questo comando non produce alcun output.

La policy è definita come documento JSON nel AdminPolicyfile.json. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, consulta Gestione delle policy IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta PutGroupPolicy in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio crea una policy in linea denominata **`AppTesterPolicy`** e la incorpora nel gruppo IAM. **`AppTesters`** Se esiste già una politica in linea con lo stesso nome, viene sovrascritta. Il contenuto della policy JSON viene fornito nel file. **`apptesterpolicy.json`** Si noti che è necessario utilizzare il **`-Raw`** parametro per elaborare correttamente il contenuto del file JSON.

```
Write-IAMGroupPolicy -GroupName AppTesters -PolicyName AppTesterPolicy -
PolicyDocument (Get-Content -Raw apptesterpolicy.json)
```

- Per i dettagli sull'API, vedere PutGroupPolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutRolePermissionsBoundary`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutRolePermissionsBoundary`.

CLI

AWS CLI

Esempio 1: applicare un limite di autorizzazioni basato su una policy personalizzata a un ruolo IAM

L'`put-role-permissions-boundary`esempio seguente applica la policy personalizzata `intern-boundary` denominata limite delle autorizzazioni per il ruolo IAM specificato.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --role-name lambda-application-role
```

Questo comando non produce alcun output.

Esempio 2: applicare un limite di autorizzazioni basato su una policy AWS gestita a un ruolo IAM

L'`put-role-permissions-boundary`esempio seguente applica la `PowerUserAccess` policy AWS gestita come limite di autorizzazioni per il ruolo IAM specificato.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --role-name x-account-admin
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta AWS CLI Command PutRolePermissionsBoundaryReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio mostra come impostare il limite di autorizzazione per un ruolo IAM. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMRolePermissionsBoundary -RoleName MyRoleName -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutRolePermissionsBoundaryin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutRolePolicy`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
```

```
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la PutRolePolicy in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::putRolePolicy(
        const Aws::String &roleName,
        const Aws::String &policyName,
        const Aws::String &policyDocument,
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::IAM::IAMClient iamClient(clientConfig);
    Aws::IAM::Model::PutRolePolicyRequest request;

    request.SetRoleName(roleName);
    request.SetPolicyName(policyName);
    request.SetPolicyDocument(policyDocument);

    Aws::IAM::Model::PutRolePolicyOutcome outcome =
 iamClient.PutRolePolicy(request);
```

```
    if (!outcome.IsSuccess()) {
        std::cerr << "Error putting policy on role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully put the role policy." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la PutRolePolicy in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come collegare una policy di autorizzazioni a un ruolo IAM

Il comando `put-role-policy` seguente aggiunge una policy di autorizzazioni al ruolo denominato `Test-Role`.

```
aws iam put-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come documento JSON nel AdminPolicyfile.json. (Il nome e l'estensione del file non hanno importanza.)

Per collegare una policy di attendibilità a un ruolo, usa il comando `update-assume-role-policy`.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta PutRolePolicy in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { PutRolePolicyCommand, IAMClient } from "@aws-sdk/client-iam";

const examplePolicyDocument = JSON.stringify({
  Version: "2012-10-17",
  Statement: [
    {
      Sid: "VisualEditor0",
      Effect: "Allow",
      Action: [
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions",
        "s3:ListBucket",
        "s3:ListMultipartUploadParts",
      ],
      Resource: "arn:aws:s3:::some-test-bucket",
    },
    {
      Sid: "VisualEditor1",
      Effect: "Allow",
      Action: [
        "s3:ListStorageLensConfigurations",
        "s3:ListAccessPointsForObjectLambda",
        "s3:ListAllMyBuckets",
        "s3:ListAccessPoints",
        "s3:ListJobs",
        "s3:ListMultiRegionAccessPoints",
      ],
      Resource: "*",
    },
  ],
});
```

```
const client = new IAMClient({});

/**
 *
 * @param {string} roleName
 * @param {string} policyName
 * @param {string} policyDocument
 */
export const putRolePolicy = async (roleName, policyName, policyDocument) => {
  const command = new PutRolePolicyCommand({
    RoleName: roleName,
    PolicyName: policyName,
    PolicyDocument: policyDocument,
  });

  const response = await client.send(command);
  console.log(response);
  return response;
};
```

- Per i dettagli sull'API, consulta la PutRolePolicy in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea una policy in linea denominata **`FedTesterRolePolicy`** e la incorpora nel ruolo IAM. **`FedTesterRole`** Se esiste già una politica in linea con lo stesso nome, viene sovrascritta. Il contenuto della policy JSON proviene dal file. **`FedTesterPolicy.json`** Si noti che è necessario utilizzare il **`-Raw`** parametro per elaborare correttamente il contenuto del file JSON.

```
Write-IAMRolePolicy -RoleName FedTesterRole -PolicyName FedTesterRolePolicy -
PolicyDocument (Get-Content -Raw FedTesterPolicy.json)
```

- Per i dettagli sull'API, vedere PutRolePolicy in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutUserPermissionsBoundary** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutUserPermissionsBoundary`.

CLI

AWS CLI

Esempio 1: applicare un limite di autorizzazioni basato su una politica personalizzata a un utente IAM

L'`put-user-permissions-boundary`esempio seguente applica una policy personalizzata `intern-boundary` denominata limite di autorizzazioni per l'utente IAM specificato.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --user-name intern
```

Questo comando non produce alcun output.

Esempio 2: applicare un limite di autorizzazioni basato su una policy AWS gestita a un utente IAM

L'`put-user-permissions-boundary`esempio seguente applica la policy AWS gestita `PowerUserAccess` denominata limite delle autorizzazioni per l'utente IAM specificato.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --user-name developer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta Command PutUserPermissionsBoundaryReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio mostra come impostare il limite di autorizzazione per l'utente. È possibile impostare politiche AWS gestite o politiche personalizzate come limite di autorizzazione.

```
Set-IAMUserPermissionsBoundary -UserName joe -PermissionsBoundary
 arn:aws:iam::123456789012:policy/intern-boundary
```

- Per i dettagli sull'API, vedere PutUserPermissionsBoundaryin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutUserPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutUserPolicy`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un utente e assunzione di un ruolo

CLI

AWS CLI

Come collegare una policy a un utente IAM

Il comando `put-user-policy` seguente collega una policy al ruolo IAM denominato `Bob`.

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come documento JSON nel AdminPolicyfile.json. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta PutUserPolicy in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(userName string, policyName string,
 actions []string,
  roleArn string) error {
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(roleArn),
```

```
 }},
}
policyBytes, err := json.Marshal(policyDoc)
if err != nil {
 log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
err)
 return err
}
_, err = wrapper.IamClient.PutUserPolicy(context.TODO(),
&iam.PutUserPolicyInput{
 PolicyDocument: aws.String(string(policyBytes)),
 PolicyName:     aws.String(policyName),
 UserName:       aws.String(userName),
})
if err != nil {
 log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
err)
}
return err
}
```

- Per i dettagli sull'API, consulta la PutUserPolicy in AWS SDK for Go API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio crea una policy in linea denominata **`EC2AccessPolicy`** e la incorpora nell'utente IAM. **`Bob`** Se esiste già una politica in linea con lo stesso nome, viene sovrascritta. Il contenuto della policy JSON proviene dal file. **`EC2AccessPolicy.json`** Si noti che è necessario utilizzare il **`-Raw`** parametro per elaborare correttamente il contenuto del file JSON.

```
Write-IAMUserPolicy -UserName Bob -PolicyName EC2AccessPolicy -PolicyDocument
 (Get-Content -Raw EC2AccessPolicy.json)
```

- Per i dettagli sull'API, vedere PutUserPolicy in AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates an inline policy for a specified user.
 # @param username [String] The name of the IAM user.
 # @param policy_name [String] The name of the policy to create.
 # @param policy_document [String] The JSON policy document.
 # @return [Boolean]
 def create_user_policy(username, policy_name, policy_document)
   @iam_client.put_user_policy({
     user_name: username,
     policy_name: policy_name,
     policy_document: policy_document
   })
   @logger.info("Policy #{policy_name} created for user #{username}.")
   true
 rescue Aws::IAM::Errors::ServiceError => e
   @logger.error("Couldn't create policy #{policy_name} for user #{username}.
Here's why:")
   @logger.error("\t#{e.code}: #{e.message}")
   false
 end
```

- Per i dettagli sull'API, consulta la PutUserPolicy in AWS SDK for Ruby API Reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   func putUserPolicy(policyDocument: String, policyName: String, user:
IAMClientTypes.User) async throws {
       let input = PutUserPolicyInput(
           policyDocument: policyDocument,
           policyName: policyName,
           userName: user.userName
       )
       do {
           _ = try await iamClient.putUserPolicy(input: input)
       } catch {
           throw error
       }
   }
```

- Per i dettagli sull'API, consulta PutUserPolicy in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RemoveClientIdFromOpenIdConnectProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveClientIdFromOpenIdConnectProvider`.

CLI

AWS CLI

Per rimuovere l'ID client specificato dall'elenco degli ID client registrati per il provider IAM OpenID Connect specificato

Questo esempio rimuove l'ID client `My-TestApp-3` dall'elenco degli ID client associati al provider IAM OIDC il cui ARN è. `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`

```
aws iam remove-client-id-from-open-id-connect-provider
    --client-id My-TestApp-3 \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta Command RemoveClientIdFromOpenIdConnectProviderReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio rimuove l'ID client **`My-TestApp-3`** dall'elenco degli ID client associati al provider IAM OIDC il cui ARN è. **`arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`**

```
Remove-IAMClientIDFromOpenIDConnectProvider -ClientID My-TestApp-3
 -OpenIDConnectProviderArn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

- Per i dettagli sull'API, vedere RemoveClientIdFromOpenIdConnectProviderin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RemoveRoleFromInstanceProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveRoleFromInstanceProfile`.

CLI

AWS CLI

Per rimuovere un ruolo da un profilo di istanza

Il `remove-role-from-instance-profile` comando seguente rimuove il ruolo denominato `Test-Role` dal profilo di istanza denominato`ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
    --instance-profile-name ExampleInstanceProfile \
    --role-name Test-Role
```

Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta RemoveRoleFromInstanceProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio elimina il ruolo denominato **`MyNewRole`** dal profilo dell'istanza EC2 denominato. **`MyNewRole`** Un profilo di istanza creato nella console IAM ha sempre lo stesso nome del ruolo, come in questo esempio. Se li crei nell'API o nella CLI, possono avere nomi diversi.

```
Remove-IAMRoleFromInstanceProfile -InstanceProfileName MyNewRole -RoleName
 MyNewRole -Force
```

- Per i dettagli sull'API, vedere RemoveRoleFromInstanceProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RemoveUserFromGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveUserFromGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione di un gruppo e aggiunta di un utente

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Remove a user from an IAM group.
   /// </summary>
   /// <param name="userName">The username of the user to remove.</param>
   /// <param name="groupName">The name of the IAM group to remove the user
from.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> RemoveUserFromGroupAsync(string userName, string
groupName)
   {
```

```
        // Remove the user from the group.
        var removeUserRequest = new RemoveUserFromGroupRequest()
        {
            UserName = userName,
            GroupName = groupName,
        };

        var response = await
_IAMService.RemoveUserFromGroupAsync(removeUserRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la RemoveUserFromGroupsezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Come rimuovere un utente da un gruppo IAM

Il comando `remove-user-from-group` seguente rimuove l'utente denominato `Bob` dal gruppo IAM denominato `Admins`.

```
aws iam remove-user-from-group \
    --user-name Bob \
    --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Aggiunta e rimozione di utenti in un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta RemoveUserFromGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio rimuove l'utente IAM **`Bob`** dal gruppo**`Testers`**.

```
Remove-IAMUserFromGroup -GroupName Testers -UserName Bob
```

Esempio 2: questo esempio trova tutti i gruppi di cui l'utente IAM **`Theresa`** è membro e quindi li rimuove **`Theresa`** da tali gruppi.

```
$groups = Get-IAMGroupForUser -UserName Theresa
foreach ($group in $groups) { Remove-IAMUserFromGroup -GroupName $group.GroupName
 -UserName Theresa -Force }
```

Esempio 3: Questo esempio mostra un modo alternativo per rimuovere l'utente IAM **`Bob`** dal **`Testers`** gruppo.

```
Get-IAMGroupForUser -UserName Bob | Remove-IAMUserFromGroup -UserName Bob -
GroupName Testers -Force
```

- Per i dettagli sull'API, vedere RemoveUserFromGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ResyncMfaDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ResyncMfaDevice`.

CLI

AWS CLI

Per sincronizzare un dispositivo MFA

L'`resync-mfa-device`esempio seguente sincronizza il dispositivo MFA associato all'Bobutente IAM e il cui ARN `arn:aws:iam::123456789012:mfa/BobsMFADevice` è con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
aws iam resync-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
```

```
    --authentication-code2 987654
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella AWS Guida per l'utente IAM.

- Per i dettagli sull'API, consulta ResyncMfa Device in Command Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio sincronizza il dispositivo MFA associato all'**Bob**utente IAM e il cui ARN **arn:aws:iam::123456789012:mfa/bob** è con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber arn:aws:iam::123456789012:mfa/theresa -
AuthenticationCode1 123456 -AuthenticationCode2 987654 -UserName Bob
```

Esempio 2: questo esempio sincronizza il dispositivo MFA IAM associato all'**Theresa**utente IAM con un dispositivo fisico che ha il **ABCD12345678** numero di serie e che ha fornito i due codici di autenticazione.

```
Sync-IAMMFADevice -SerialNumber ABCD12345678 -AuthenticationCode1 123456 -
AuthenticationCode2 987654 -UserName Theresa
```

- Per i dettagli sull'API, vedere ResyncMfaDevice in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **SetDefaultPolicyVersion** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`SetDefaultPolicyVersion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Gestione delle policy
- Rollback di una versione della policy

CLI

AWS CLI

Per impostare la versione specificata della politica specificata come versione predefinita della politica.

Questo esempio imposta la v2 versione della policy il cui ARN è `arn:aws:iam::123456789012:policy/MyPolicy` la versione attiva predefinita.

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, vedere SetDefaultPolicyVersionin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio imposta la **v2** versione della policy il cui ARN è **`arn:aws:iam::123456789012:policy/MyPolicy`** la versione attiva predefinita.

```
Set-IAMDefaultPolicyVersion -PolicyArn arn:aws:iam::123456789012:policy/MyPolicy
 -VersionId v2
```

- Per i dettagli sull'API, vedere SetDefaultPolicyVersionin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`TagRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`TagRole`.

CLI

AWS CLI

Per aggiungere un tag a un ruolo

Il `tag-role` comando seguente aggiunge un tag con un nome di reparto al ruolo specificato.

```
aws iam tag-role --role-name my-role \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta TagRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiunge un tag a Role in Identity Management Service

```
Add-IAMRoleTag -RoleName AdminRoleacess -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagRolein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`TagUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`TagUser`.

CLI

AWS CLI

Per aggiungere un tag a un utente

Il `tag-user` comando seguente aggiunge un tag con il Dipartimento associato all'utente specificato.

```
aws iam tag-user \
    --user-name alice \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta TagUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiunge un tag a User in Identity Management Service

```
Add-IAMUserTag -UserName joe -Tag @{ Key = 'abac'; Value = 'testing'}
```

- Per i dettagli sull'API, vedere TagUserin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UntagRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UntagRole`.

CLI

AWS CLI

Per rimuovere un tag da un ruolo

Il `untag-role` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal ruolo specificato.

```
aws iam untag-role \
    --role-name my-role \
    --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta UntagRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio rimuove il tag dal ruolo denominato "MyRoleNome» con la chiave del tag «abac». Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMRoleTag -RoleName MyRoleName -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagRolein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UntagUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UntagUser`.

CLI

AWS CLI

Per rimuovere un tag da un utente

Il `untag-user` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dall'utente specificato.

```
aws iam untag-user \
    --user-name alice \
    --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Tagging IAM resources nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta UntagUser AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio rimuove il tag dall'utente chiamato «joe» con la chiave del tag «abac» e «xyzw». Per rimuovere più tag, fornisci un elenco di chiavi di tag separate da virgole.

```
Remove-IAMUserTag -UserName joe -TagKey "abac","xyzw"
```

- Per i dettagli sull'API, vedere UntagUserin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateAccessKey`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAccessKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestione delle chiavi di accesso

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateAccessKey(const Aws::String &userName,
                                  const Aws::String &accessKeyID,
                                  Aws::IAM::Model::StatusType status,
                                  const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::UpdateAccessKeyRequest request;
    request.SetUserName(userName);
    request.SetAccessKeyId(accessKeyID);
    request.SetStatus(status);

    auto outcome = iam.UpdateAccessKey(request);
    if (outcome.IsSuccess()) {
        std::cout << "Successfully updated status of access key "
                  << accessKeyID << " for user " << userName << std::endl;
    }
    else {
        std::cerr << "Error updated status of access key " << accessKeyID <<
                  " for user " << userName << ": " <<
                  outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta UpdateAccessKey in AWS SDK for C++ API Reference.

CLI

AWS CLI

Come attivare o disattivare una chiave di accesso per un utente IAM

Il comando `update-access-key` seguente disattiva la chiave di accesso specificata (ID chiave di accesso e chiave di accesso segreta) per l'utente IAM denominato Bob.

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Questo comando non produce alcun output.

La disattivazione della chiave significa che non può essere utilizzata per l'accesso programmatico a. AWS La chiave, tuttavia, rimane disponibile e può essere riattivata.

Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateAccessKey in AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.StatusType;
import software.amazon.awssdk.services.iam.model.UpdateAccessKeyRequest;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateAccessKey {

    private static StatusType statusType;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <username> <accessId> <status>\s

                Where:
                    username - The name of the user whose key you want to update.
\s
                    accessId - The access key ID of the secret access key you
 want to update.\s
                    status - The status you want to assign to the secret access
 key.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String username = args[0];
        String accessId = args[1];
        String status = args[2];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        updateKey(iam, username, accessId, status);
        System.out.println("Done");
        iam.close();
    }
```

```
    public static void updateKey(IamClient iam, String username, String accessId,
 String status) {
        try {
            if (status.toLowerCase().equalsIgnoreCase("active")) {
                statusType = StatusType.ACTIVE;
            } else if (status.toLowerCase().equalsIgnoreCase("inactive")) {
                statusType = StatusType.INACTIVE;
            } else {
                statusType = StatusType.UNKNOWN_TO_SDK_VERSION;
            }

            UpdateAccessKeyRequest request = UpdateAccessKeyRequest.builder()
                    .accessKeyId(accessId)
                    .userName(username)
                    .status(statusType)
                    .build();

            iam.updateAccessKey(request);
            System.out.printf("Successfully updated the status of access key %s
 to" +
                    "status %s for user %s", accessId, status, username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta UpdateAccessKey in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiorna la chiave di accesso.

```
import {
  UpdateAccessKeyCommand,
  IAMClient,
  StatusType,
} from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} userName
 * @param {string} accessKeyId
 */
export const updateAccessKey = (userName, accessKeyId) => {
  const command = new UpdateAccessKeyCommand({
    AccessKeyId: accessKeyId,
    Status: StatusType.Inactive,
    UserName: userName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta UpdateAccessKey in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
```

```
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  AccessKeyId: "ACCESS_KEY_ID",
  Status: "Active",
  UserName: "USER_NAME",
};

iam.updateAccessKey(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta UpdateAccessKey in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio modifica lo stato della chiave di accesso **AKIAIOSFODNN7EXAMPLE** per l'utente IAM denominato **Bob** to**Inactive**.

```
Update-IAMAccessKey -UserName Bob -AccessKeyId AKIAIOSFODNN7EXAMPLE -Status
 Inactive
```

- Per i dettagli sull'API, vedere UpdateAccessKey in AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def update_key(user_name, key_id, activate):
    """
    Updates the status of a key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to update.
    :param activate: When True, the key is activated. Otherwise, the key is
 deactivated.
    """

    try:
        key = iam.User(user_name).AccessKey(key_id)
        if activate:
            key.activate()
        else:
            key.deactivate()
        logger.info("%s key %s.", "Activated" if activate else "Deactivated",
 key_id)
    except ClientError:
        logger.exception(
            "Couldn't %s key %s.", "Activate" if activate else "Deactivate",
 key_id
        )
        raise
```

- Per i dettagli sull'API, consulta UpdateAccessKey in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateAccountPasswordPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAccountPasswordPolicy`.

CLI

AWS CLI

Per impostare o modificare la politica corrente in materia di password dell'account

Il `update-account-password-policy` comando seguente imposta la politica delle password in modo che richieda una lunghezza minima di otto caratteri e richieda uno o più numeri nella password.

```
aws iam update-account-password-policy \
    --minimum-password-length 8 \
    --require-numbers
```

Questo comando non produce alcun output.

Le modifiche alla politica in materia di password di un account influiscono su tutte le nuove password create per gli utenti IAM nell'account. Le modifiche alle politiche relative alle password non influiscono sulle password esistenti.

Per ulteriori informazioni, consulta Impostazione di una policy delle password dell'account per utenti IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateAccountPasswordPolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna la politica delle password per l'account con le impostazioni specificate. Si noti che tutti i parametri non inclusi nel comando non vengono lasciati invariati. Vengono invece ripristinati ai valori predefiniti.

```
Update-IAMAccountPasswordPolicy -AllowUsersToChangePasswords $true -HardExpiry
 $false -MaxPasswordAge 90 -MinimumPasswordLength 8 -PasswordReusePrevention 20
 -RequireLowercaseCharacters $true -RequireNumbers $true -RequireSymbols $true -
RequireUppercaseCharacters $true
```

- Per i dettagli sull'API, vedere UpdateAccountPasswordPolicyin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateAssumeRolePolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAssumeRolePolicy`.

CLI

AWS CLI

Per aggiornare la policy di fiducia per un ruolo IAM

Il `update-assume-role-policy` comando seguente aggiorna la politica di fiducia per il ruolo denominato`Test-Role`.

```
aws iam update-assume-role-policy \
    --role-name Test-Role \
    --policy-document file://Test-Role-Trust-Policy.json
```

Questo comando non produce alcun output.

La policy di attendibilità è definita come documento JSON nel file Test-Role-Trust-Policy.json. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per aggiornare la politica delle autorizzazioni per un ruolo, usa il `put-role-policy` comando.

Per ulteriori informazioni, consulta Creazione di ruoli IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateAssumeRolePolicy AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna il ruolo IAM denominato **`ClientRole`** con una nuova policy di fiducia, il cui contenuto proviene dal file**`ClientRolePolicy.json`**. Si noti che è necessario utilizzare il parametro **`-Raw`** switch per elaborare correttamente il contenuto del file JSON.

```
Update-IAMAssumeRolePolicy -RoleName ClientRole -PolicyDocument (Get-Content -raw
 ClientRolePolicy.json)
```

- Per i dettagli sull'API, vedere UpdateAssumeRolePolicyin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateGroup`.

CLI

AWS CLI

Per rinominare un gruppo IAM

Il `update-group` comando seguente modifica il nome del gruppo IAM `Test` in`Test-1`.

```
aws iam update-group \
    --group-name Test \
    --new-group-name Test-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Ridenominazione di un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateGroup AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio rinomina il gruppo IAM **`Testers`** in**`AppTesters`**.

```
Update-IAMGroup -GroupName Testers -NewGroupName AppTesters
```

Esempio 2: Questo esempio modifica il percorso del gruppo IAM **`AppTesters`** in**`/Org1/Org2/`**. Questo modifica l'ARN del gruppo in. **`arn:aws:iam::123456789012:group/Org1/Org2/AppTesters`**

```
Update-IAMGroup -GroupName AppTesters -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateGroupin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateLoginProfile`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateLoginProfile`.

CLI

AWS CLI

Per aggiornare la password di un utente IAM

Il `update-login-profile` comando seguente crea una nuova password per l'utente IAM denominatoBob.

```
aws iam update-login-profile \
    --user-name Bob \
    --password <password>
```

Questo comando non produce alcun output.

Per impostare una politica di password per l'account, usa il `update-account-password-policy` comando. Se la nuova password viola la politica relativa alle password dell'account, il comando restituisce un `PasswordPolicyViolation` errore.

Se la politica sulla password dell'account lo consente, gli utenti IAM possono modificare le proprie password utilizzando il `change-password` comando.

Conserva la password in un luogo sicuro. Se la password viene persa, non può essere recuperata ed è necessario crearne una nuova utilizzando il `create-login-profile` comando.

Per ulteriori informazioni, consulta Managing password for IAM users nella AWS IAM User Guide.

- Per i dettagli sull'API, consulta UpdateLoginProfile in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio imposta una nuova password temporanea per l'utente **Bob** IAM e richiede all'utente di modificare la password al successivo accesso.

```
Update-IAMLoginProfile -UserName Bob -Password "P@ssw0rd1234" -
PasswordResetRequired $true
```

- Per i dettagli sull'API, vedere UpdateLoginProfile in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateOpenIdConnectProviderThumbprint`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateOpenIdConnectProviderThumbprint`.

CLI

AWS CLI

Per sostituire l'elenco esistente di impronte digitali dei certificati del server con un nuovo elenco

Questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il provider OIDC il cui ARN deve utilizzare una nuova impronta `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` personale.

```
aws iam update-open-id-connect-provider-thumbprint \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com \
    --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Creating OpenID Connect (OIDC) di Identity Provider nella IAM User Guide.AWS

- Per i dettagli sulle API, consulta Command UpdateOpenIdConnectProviderThumbprintReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il provider OIDC il cui ARN deve utilizzare una nuova impronta **`arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`** personale. Il provider OIDC condivide il nuovo valore quando il certificato associato al provider cambia.

```
Update-IAMOpenIDConnectProviderThumbprint -OpenIDConnectProviderArn
 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com -ThumbprintList
 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

- Per i dettagli sull'API, vedere UpdateOpenIdConnectProviderThumbprintin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateRole`.

CLI

AWS CLI

Per modificare la descrizione o la durata della sessione di un ruolo IAM

Il `update-role` comando seguente modifica la descrizione del ruolo IAM `production-role` in `Main production role` e imposta la durata massima della sessione su 12 ore.

```
aws iam update-role \
    --role-name production-role \
    --description 'Main production role' \
    --max-session-duration 43200
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateRole AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna la descrizione del ruolo e il valore della durata massima della sessione (in secondi) per cui è possibile richiedere la sessione di un ruolo.

```
Update-IAMRole -RoleName MyRoleName -Description "My testing role" -
MaxSessionDuration 43200
```

- Per i dettagli sull'API, vedere UpdateRolein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateRoleDescription`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateRoleDescription`.

CLI

AWS CLI

Per modificare la descrizione di un ruolo IAM

Il `update-role` comando seguente modifica la descrizione del ruolo IAM `production-role` in`Main production role`.

```
aws iam update-role-description \
    --role-name production-role \
    --description 'Main production role'
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "production-role",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/production-role",
        "CreateDate": "2017-12-06T17:16:37+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {}
                }
            ]
        },
```

```
        "Description": "Main production role"
    }
}
```

Per ulteriori informazioni, consulta Modifica di un ruolo nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta la UpdateRoledescrizione in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna la descrizione di un ruolo IAM nel tuo account.

```
Update-IAMRoleDescription -RoleName MyRoleName -Description "My testing role"
```

- Per i dettagli sull'API, consulta la UpdateRoledescrizione in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateSamlProvider`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateSamlProvider`.

CLI

AWS CLI

Per aggiornare il documento di metadati per un provider SAML esistente

Questo esempio aggiorna il provider SAML in IAM il cui ARN `arn:aws:iam::123456789012:saml-provider/SAMLADFS` è con un nuovo documento di metadati SAML dal file. `SAMLMetaData.xml`

```
aws iam update-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Output:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

Per ulteriori informazioni, consulta Creazione di provider di identità SAML IAM nella Guida per l'utente di IAM AWS .

- Per i dettagli sull'API, consulta UpdateSamlProvider in Command Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna il provider SAML in IAM il cui ARN **`arn:aws:iam::123456789012:saml-provider/SAMLADFS`** è con un nuovo documento di metadati SAML dal file. **`SAMLMetaData.xml`** Tieni presente che devi utilizzare il parametro **`-Raw`** switch per elaborare correttamente il contenuto del file JSON.

```
Update-IAMSAMLProvider -SAMLProviderArn arn:aws:iam::123456789012:saml-provider/
SAMLADFS -SAMLMetadataDocument (Get-Content -Raw SAMLMetaData.xml)
```

- Per i dettagli sull'API, vedere UpdateSamlProvider in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateServerCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateServerCertificate`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateServerCertificate(const Aws::String
 &currentCertificateName,
                                           const Aws::String &newCertificateName,
                                           const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);
    Aws::IAM::Model::UpdateServerCertificateRequest request;
    request.SetServerCertificateName(currentCertificateName);
    request.SetNewServerCertificateName(newCertificateName);

    auto outcome = iam.UpdateServerCertificate(request);
    bool result = true;
    if (outcome.IsSuccess()) {
        std::cout << "Server certificate " << currentCertificateName
                  << " successfully renamed as " << newCertificateName
                  << std::endl;
    }
    else {
        if (outcome.GetError().GetErrorType() !=
 Aws::IAM::IAMErrors::NO_SUCH_ENTITY) {
            std::cerr << "Error changing name of server certificate " <<
                      currentCertificateName << " to " << newCertificateName <<
 ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Certificate '" << currentCertificateName
                      << "' not found." << std::endl;
        }
    }
```

```
    return result;
}
```

- Per i dettagli sull'API, consulta UpdateServerCertificate in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per modificare il percorso o il nome di un certificato del server nel tuo AWS account

Il comando `update-server-certificate` seguente modifica il nome del certificato da `myServerCertificate` a `myUpdatedServerCertificate`. Cambia anche il percorso in `/cloudfront/` modo che sia possibile accedervi dal CloudFront servizio Amazon. Questo comando non produce alcun output. Puoi visualizzare i risultati dell'aggiornamento eseguendo il comando `list-server-certificates`.

```
aws-iam update-server-certificate \
    --server-certificate-name myServerCertificate \
    --new-server-certificate-name myUpdatedServerCertificate \
    --new-path /cloudfront/
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Gestione dei certificati server in IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateServerCertificate in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiorna un certificato del server.

```
import { UpdateServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} currentName
 * @param {string} newName
 */
export const updateServerCertificate = (currentName, newName) => {
  const command = new UpdateServerCertificateCommand({
    ServerCertificateName: currentName,
    NewServerCertificateName: newName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta UpdateServerCertificate in AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  ServerCertificateName: "CERTIFICATE_NAME",
```

```
  NewServerCertificateName: "NEW_CERTIFICATE_NAME",
};

iam.updateServerCertificate(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta UpdateServerCertificate in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio rinomina il certificato **MyServerCertificate** denominato**MyRenamedServerCertificate**.

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -
NewServerCertificateName MyRenamedServerCertificate
```

Esempio 2: Questo esempio sposta il certificato denominato nel percorso **MyServerCertificate** /Org1/Org2/. Questo modifica l'ARN della risorsa in. **arn:aws:iam::123456789012:server-certificate/Org1/Org2/MyServerCertificate**

```
Update-IAMServerCertificate -ServerCertificateName MyServerCertificate -NewPath /
Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateServerCertificate in AWS Tools for PowerShell Cmdlet Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca, aggiorna ed elimina i certificati del server.

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
                                            server_certificate_name: name,
                                            certificate_body: certificate_body,
                                            private_key: private_key,
                                          })
    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info("No server certificates found.")
      return
```

```
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
      @logger.info("Certificate Name:
 #{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
 '#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
```

- Per i dettagli sull'API, consulta UpdateServerCertificate in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateSigningCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateSigningCertificate`.

CLI

AWS CLI

Per attivare o disattivare un certificato di firma per un utente IAM

Il `update-signing-certificate` comando seguente disattiva il certificato di firma specificato per l'utente IAM denominato. `Bob`

```
aws iam update-signing-certificate \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

Per ottenere l'ID per un certificato di firma, usa il `list-signing-certificates` comando.

Per ulteriori informazioni, consulta Gestire i certificati di firma nella Guida per l'utente di Amazon EC2.

- Per i dettagli sull'API, consulta UpdateSigningCertificate in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio aggiorna il certificato associato all'utente IAM denominato **`Bob`** e il cui ID di certificato serve **`Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU`** a contrassegnarlo come inattivo.

```
Update-IAMSigningCertificate -CertificateId Y3EK7RMEXAMPLESV33FCREXAMPLEMJLU -
UserName Bob -Status Inactive
```

- Per i dettagli sull'API, vedere UpdateSigningCertificate in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Creazione di utenti di sola lettura e di lettura e scrittura

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::IAM::updateUser(const Aws::String &currentUserName,
                             const Aws::String &newUserName,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::IAM::IAMClient iam(clientConfig);

    Aws::IAM::Model::UpdateUserRequest request;
    request.SetUserName(currentUserName);
    request.SetNewUserName(newUserName);

    auto outcome = iam.UpdateUser(request);
    if (outcome.IsSuccess()) {
        std::cout << "IAM user " << currentUserName <<
                  " successfully updated with new user name " << newUserName <<
                  std::endl;
    }
    else {
```

```
        std::cerr << "Error updating user name for IAM user " << currentUserName
 <<
                  ":" << outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Come modificare il nome di un utente IAM

Il comando `update-user` seguente modifica il nome di un utente IAM da `Bob` a `Robert`.

```
aws iam update-user \
    --user-name Bob \
    --new-user-name Robert
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Ridenominazione di un gruppo di utenti IAM nella Guida per l'utente IAM AWS .

- Per i dettagli sull'API, consulta UpdateUser AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.IamException;
import software.amazon.awssdk.services.iam.model.UpdateUserRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateUser {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <curName> <newName>\s

                Where:
                    curName - The current user name.\s
                    newName - An updated user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String curName = args[0];
        String newName = args[1];
        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        updateIAMUser(iam, curName, newName);
        System.out.println("Done");
        iam.close();
    }

    public static void updateIAMUser(IamClient iam, String curName, String
 newName) {
```

```
        try {
            UpdateUserRequest request = UpdateUserRequest.builder()
                    .userName(curName)
                    .newUserName(newName)
                    .build();

            iam.updateUser(request);
            System.out.printf("Successfully updated user to username %s",
 newName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiorna l'utente.

```
import { UpdateUserCommand, IAMClient } from "@aws-sdk/client-iam";

const client = new IAMClient({});

/**
 *
 * @param {string} currentUserName
 * @param {string} newUserName
 */
export const updateUser = (currentUserName, newUserName) => {
```

```
  const command = new UpdateUserCommand({
    UserName: currentUserName,
    NewUserName: newUserName,
  });

  return client.send(command);
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the IAM service object
var iam = new AWS.IAM({ apiVersion: "2010-05-08" });

var params = {
  UserName: process.argv[2],
  NewUserName: process.argv[3],
};

iam.updateUser(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun updateIAMUser(
    curName: String?,
    newName: String?
) {
    val request =
        UpdateUserRequest {
            userName = curName
            newUserName = newName
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        iamClient.updateUser(request)
        println("Successfully updated user to $newName")
    }
}
```

- Per i dettagli sull'API, UpdateUserconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio rinomina l'utente IAM **Bob** in**Robert**.

```
Update-IAMUser -UserName Bob -NewUserName Robert
```

Esempio 2: questo esempio modifica il percorso dell'utente IAM **Bob** in**/Org1/Org2/**, il che modifica effettivamente l'ARN dell'utente in. **arn:aws:iam::123456789012:user/Org1/Org2/bob**

```
Update-IAMUser -UserName Bob -NewPath /Org1/Org2/
```

- Per i dettagli sull'API, vedere UpdateUserin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def update_user(user_name, new_user_name):
    """
    Updates a user's name.

    :param user_name: The current name of the user to update.
    :param new_user_name: The new name to assign to the user.
    :return: The updated user.
    """
    try:
        user = iam.User(user_name)
        user.update(NewUserName=new_user_name)
        logger.info("Renamed %s to %s.", user_name, new_user_name)
    except ClientError:
        logger.exception("Couldn't update name for user %s.", user_name)
        raise
    return user
```

- Per i dettagli sull'API, consulta UpdateUser AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Updates an IAM user's name
#
# @param current_name [String] The current name of the user
# @param new_name [String] The new name of the user
def update_user_name(current_name, new_name)
  @iam_client.update_user(user_name: current_name, new_user_name: new_name)
  true
rescue StandardError => e
  @logger.error("Error updating user name from '#{current_name}' to
'#{new_name}': #{e.message}")
  false
end
```

- Per i dettagli sull'API, consulta la UpdateUsersezione AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UploadServerCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UploadServerCertificate`.

CLI

AWS CLI

Per caricare un certificato server sul tuo account AWS

Il seguente comando upload-server-certificate carica un certificato server sul tuo account. AWS In questo esempio, il certificato è nel file `public_key_cert_file.pem`, la chiave privata associata è nel file `my_private_key.pem` e la catena di certificati fornita dall'autorità di certificazione (CA) è nel file `my_certificate_chain_file.pem`. Al termine del caricamento, il file è disponibile con il nome my. ServerCertificate I parametri che iniziano con `file://` indicano al comando di leggere il contenuto del file e di utilizzarlo come valore del parametro in luogo del nome del file.

```
aws iam upload-server-certificate \
    --server-certificate-name myServerCertificate \
    --certificate-body file://public_key_cert_file.pem \
    --private-key file://my_private_key.pem \
    --certificate-chain file://my_certificate_chain_file.pem
```

Output:

```
{
    "ServerCertificateMetadata": {
        "Path": "/",
        "ServerCertificateName": "myServerCertificate",
        "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
        "Arn": "arn:aws:iam::1234567989012:server-certificate/
myServerCertificate",
        "UploadDate": "2019-04-22T21:13:44+00:00",
        "Expiration": "2019-10-15T22:23:16+00:00"
    }
}
```

Per ulteriori informazioni, consulta Creazione, caricamento ed eliminazione di certificati server nella guida Utilizzo di IAM

- Per i dettagli sull'API, consulta UploadServerCertificate in AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { UploadServerCertificateCommand, IAMClient } from "@aws-sdk/client-iam";
import { readFileSync } from "fs";
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";
import * as path from "path";

const client = new IAMClient({});

const certMessage = `Generate a certificate and key with the following command,
 or the equivalent for your system.

openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 -nodes \
-keyout example.key -out example.crt -subj "/CN=example.com" \
-addext "subjectAltName=DNS:example.com,DNS:www.example.net,IP:10.0.0.1"
`;

const getCertAndKey = () => {
  try {
    const cert = readFileSync(
      path.join(dirnameFromMetaUrl(import.meta.url), "./example.crt"),
    );
    const key = readFileSync(
      path.join(dirnameFromMetaUrl(import.meta.url), "./example.key"),
    );
    return { cert, key };
  } catch (err) {
    if (err.code === "ENOENT") {
      throw new Error(
        `Certificate and/or private key not found. ${certMessage}`,
      );
    }

    throw err;
```

```
  }
};

/**
 *
 * @param {string} certificateName
 */
export const uploadServerCertificate = (certificateName) => {
  const { cert, key } = getCertAndKey();
  const command = new UploadServerCertificateCommand({
    ServerCertificateName: certificateName,
    CertificateBody: cert.toString(),
    PrivateKey: key.toString(),
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta UploadServerCertificate in AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio carica un nuovo certificato del server sull'account IAM. I file contenenti il corpo del certificato, la chiave privata e (facoltativamente) la catena di certificati devono essere tutti codificati in PEM. Nota che i parametri richiedono il contenuto effettivo dei file anziché i nomi dei file. È necessario utilizzare il parametro **`-Raw`** switch per elaborare correttamente il contenuto del file.

```
Publish-IAMServerCertificate -ServerCertificateName MyTestCert -CertificateBody
 (Get-Content -Raw server.crt) -PrivateKey (Get-Content -Raw server.key)
```

Output:

```
Arn                   : arn:aws:iam::123456789012:server-certificate/MyTestCert
Expiration            : 1/14/2018 9:52:36 AM
Path                  : /
ServerCertificateId   : ASCAJIEXAMPLE7J7HQZYW
```

```
ServerCertificateName : MyTestCert
UploadDate            : 4/21/2015 11:14:16 AM
```

- Per i dettagli sull'API, vedere UploadServerCertificate in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UploadSigningCertificate`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UploadSigningCertificate`.

CLI

AWS CLI

Per caricare un certificato di firma per un utente IAM

Il `upload-signing-certificate` comando seguente carica un certificato di firma per l'utente IAM denominatoBob.

```
aws iam upload-signing-certificate \
    --user-name Bob \
    --certificate-body file://certificate.pem
```

Output:

```
{
    "Certificate": {
        "UserName": "Bob",
        "Status": "Active",
        "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
 CERTIFICATE-----",
        "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
        "UploadDate": "2013-06-06T21:40:08.121Z"
    }
}
```

Il certificato si trova in un file denominato certificate.pem in formato PEM.

Per ulteriori informazioni, consulta Creazione e caricamento di un certificato di firma utente nella guida all'utilizzo di IAM.

- Per i dettagli sull'API, consulta UploadSigningCertificate in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio carica un nuovo certificato di firma X.509 e lo associa all'utente IAM denominato. **`Bob`** Il file contenente il corpo del certificato è codificato in PEM. Il **`CertificateBody`** parametro richiede il contenuto effettivo del file di certificato anziché il nome del file. È necessario utilizzare il parametro **`-Raw`** switch per elaborare correttamente il file.

```
Publish-IAMSigningCertificate -UserName Bob -CertificateBody (Get-Content -Raw
 SampleSigningCert.pem)
```

Output:

```
CertificateBody : -----BEGIN CERTIFICATE-----

 MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC

 VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6

 b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd

 BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN

 MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD

 VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z

 b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
                   YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn
+a4GmWIWJ
                   21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9T
                   rDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
```

```
 Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
                  nUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb

 FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
                  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
                  -----END CERTIFICATE-----
CertificateId   : Y3EK7RMEXAMPLESV33FCEXAMPLEHMJLU
Status          : Active
UploadDate      : 4/20/2015 1:26:01 PM
UserName        : Bob
```

- Per i dettagli sull'API, vedere UploadSigningCertificate in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per IAM che utilizzano AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni in IAM con AWS SDK. Questi scenari illustrano come eseguire attività specifiche richiamando più funzioni in IAM. Ogni scenario include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Crea e gestisci un servizio resiliente utilizzando un SDK AWS
- Crea un gruppo IAM e aggiungi un utente al gruppo utilizzando un AWS SDK
- Crea un utente IAM e assumi un ruolo AWS STS utilizzando un AWS SDK
- Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un SDK AWS
- Gestisci le chiavi di accesso IAM utilizzando un AWS SDK
- Gestisci le policy IAM utilizzando un AWS SDK
- Gestisci i ruoli IAM utilizzando un AWS SDK
- Gestisci il tuo account IAM utilizzando un AWS SDK
- Ripristina una versione della policy IAM utilizzando un AWS SDK

- Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

Crea e gestisci un servizio resiliente utilizzando un SDK AWS

I seguenti esempi di codice mostrano come creare un servizio web con bilanciamento del carico che restituisca consigli su libri, film e canzoni. L'esempio mostra come il servizio risponde ai guasti e spiega come ristrutturarlo per una maggiore resilienza in caso di guasti.

- Utilizza un gruppo con dimensionamento automatico Amazon EC2 per creare istanze Amazon Elastic Compute Cloud (Amazon EC2) basate su un modello di avvio e per mantenere il numero di istanze entro un intervallo specificato.
- Gestisci e distribuisci le richieste HTTP con Elastic Load Balancing.
- Monitora lo stato delle istanze in un gruppo con dimensionamento automatico e inoltra le richieste soltanto alle istanze integre.
- Esegui un server Web Python su ogni istanza EC2 per gestire le richieste HTTP. Il server Web risponde con consigli e controlli dell'integrità.
- Simula un servizio di raccomandazione con una tabella Amazon DynamoDB.
- Controlla la risposta del server web alle richieste e ai controlli di integrità aggiornando AWS Systems Manager i parametri.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
    static async Task Main(string[] args)
    {
        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
```

```
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();


        // Set up dependency injection for the AWS services.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonIdentityManagementService>()
                    .AddAWSService<IAmazonDynamoDB>()
                    .AddAWSService<IAmazonElasticLoadBalancingV2>()
                    .AddAWSService<IAmazonSimpleSystemsManagement>()
                    .AddAWSService<IAmazonAutoScaling>()
                    .AddAWSService<IAmazonEC2>()
                    .AddTransient<AutoScalerWrapper>()
                    .AddTransient<ElasticLoadBalancerWrapper>()
                    .AddTransient<SmParameterWrapper>()
                    .AddTransient<Recommendations>()
                    .AddSingleton<IConfiguration>(_configuration)
            )
            .Build();

        ServicesSetup(host);
        ResourcesSetup();

        try
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Welcome to the Resilient Architecture Example
Scenario.");
            Console.WriteLine(new string('-', 80));
            await Deploy(true);

            Console.WriteLine("Now let's begin the scenario.");
            Console.WriteLine(new string('-', 80));
            await Demo(true);

            Console.WriteLine(new string('-', 80));
```

```
            Console.WriteLine("Finally, let's clean up our resources.");
            Console.WriteLine(new string('-', 80));

            await DestroyResources(true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Resilient Architecture Example Scenario is
complete.");
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine($"There was a problem running the scenario:
{ex.Message}");
            await DestroyResources(true);
            Console.WriteLine(new string('-', 80));
        }
    }

    /// <summary>
    /// Setup any common resources, also used for integration testing.
    /// </summary>
    public static void ResourcesSetup()
    {
        _httpClient = new HttpClient();
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _elasticLoadBalancerWrapper =
host.Services.GetRequiredService<ElasticLoadBalancerWrapper>();
        _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
        _recommendations = host.Services.GetRequiredService<Recommendations>();
        _autoScalerWrapper =
host.Services.GetRequiredService<AutoScalerWrapper>();
        _smParameterWrapper =
host.Services.GetRequiredService<SmParameterWrapper>();
    }
```

```
    /// <summary>
    /// Deploy necessary resources for the scenario.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Deploy(bool interactive)
    {
        var protocol = "HTTP";
        var port = 80;
        var sshPort = 22;

        Console.WriteLine(
            "\nFor this demo, we'll use the AWS SDK for .NET to create several
AWS resources\n" +
            "to set up a load-balanced web service endpoint and explore some ways
to make it resilient\n" +
            "against various kinds of failures.\n\n" +
            "Some of the resources create by this demo are:\n");

        Console.WriteLine(
            "\t* A DynamoDB table that the web service depends on to provide
book, movie, and song recommendations.");
        Console.WriteLine(
            "\t* An EC2 launch template that defines EC2 instances that each
contain a Python web server.");
        Console.WriteLine(
            "\t* An EC2 Auto Scaling group that manages EC2 instances across
several Availability Zones.");
        Console.WriteLine(
            "\t* An Elastic Load Balancing (ELB) load balancer that targets the
Auto Scaling group to distribute requests.");
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to start deploying
resources.");
        if (interactive)
            Console.ReadLine();

        // Create and populate the DynamoDB table.
        var databaseTableName = _configuration["databaseName"];
        var recommendationsPath = Path.Join(_configuration["resourcePath"],
            "recommendations_objects.json");
        Console.WriteLine($"Creating and populating a DynamoDB table named
{databaseTableName}.");
```

```
        await _recommendations.CreateDatabaseWithName(databaseTableName);
        await _recommendations.PopulateDatabase(databaseTableName,
recommendationsPath);
        Console.WriteLine(new string('-', 80));

        // Create the EC2 Launch Template.

        Console.WriteLine(
            $"Creating an EC2 launch template that runs
'server_startup_script.sh' when an instance starts.\n"
            + "\nThis script starts a Python web server defined in the
`server.py` script. The web server\n"
            + "listens to HTTP requests on port 80 and responds to requests to
'/' and to '/healthcheck'.\n"
            + "For demo purposes, this server is run as the root user. In
production, the best practice is to\n"
            + "run a web server, such as Apache, with least-privileged
credentials.");
        Console.WriteLine(
            "\nThe template also defines an IAM policy that each instance uses to
assume a role that grants\n"
            + "permissions to access the DynamoDB recommendation table and
Systems Manager parameters\n"
            + "that control the flow of the demo.");

        var startupScriptPath = Path.Join(_configuration["resourcePath"],
            "server_startup_script.sh");
        var instancePolicyPath = Path.Join(_configuration["resourcePath"],
            "instance_policy.json");
        await _autoScalerWrapper.CreateTemplate(startupScriptPath,
instancePolicyPath);
        Console.WriteLine(new string('-', 80));

        Console.WriteLine(
            "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different\n"
            + "Availability Zone.\n");
        var zones = await _autoScalerWrapper.DescribeAvailabilityZones();
        await _autoScalerWrapper.CreateGroupOfSize(3,
_autoScalerWrapper.GroupName, zones);
        Console.WriteLine(new string('-', 80));

        Console.WriteLine(
```

```
            "At this point, you have EC2 instances created. Once each instance
starts, it listens for\n"
            + "HTTP requests. You can see these instances in the console or
continue with the demo.\n");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("Creating variables that control the flow of the
demo.");
        await _smParameterWrapper.Reset();

        Console.WriteLine(
            "\nCreating an Elastic Load Balancing target group and load balancer.
The target group\n"
            + "defines how the load balancer connects to instances. The load
balancer provides a\n"
            + "single endpoint where clients connect and dispatches requests to
instances in the group.");

        var defaultVpc = await _autoScalerWrapper.GetDefaultVpc();
        var subnets = await
_autoScalerWrapper.GetAllVpcSubnetsForZones(defaultVpc.VpcId, zones);
        var subnetIds = subnets.Select(s => s.SubnetId).ToList();
        var targetGroup = await
_elasticLoadBalancerWrapper.CreateTargetGroupOnVpc(_elasticLoadBalancerWrapper.TargetGro
protocol, port, defaultVpc.VpcId);

        await
_elasticLoadBalancerWrapper.CreateLoadBalancerAndListener(_elasticLoadBalancerWrapper.Lo
subnetIds, targetGroup);
        await
_autoScalerWrapper.AttachLoadBalancerToGroup(_autoScalerWrapper.GroupName,
targetGroup.TargetGroupArn);
        Console.WriteLine("\nVerifying access to the load balancer endpoint...");
        var endPoint = await
_elasticLoadBalancerWrapper.GetEndpointForLoadBalancerByName(_elasticLoadBalancerWrapper
        var loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);

        if (!loadBalancerAccess)
        {
```

```
            Console.WriteLine("\nCouldn't connect to the load balancer, verifying
 that the port is open...");

            var ipString = await _httpClient.GetStringAsync("https://
checkip.amazonaws.com");
            ipString = ipString.Trim();

            var defaultSecurityGroup = await
 _autoScalerWrapper.GetDefaultSecurityGroupForVpc(defaultVpc);
            var portIsOpen =
 _autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, port,
 ipString);
            var sshPortIsOpen =
 _autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, sshPort,
 ipString);

            if (!portIsOpen)
            {
                Console.WriteLine(
                    "\nFor this example to work, the default security group for
 your default VPC must\n"
                    + "allows access from this computer. You can either add it
 automatically from this\n"
                    + "example or add it yourself using the AWS Management
 Console.\n");

                if (!interactive || GetYesNoResponse(
                        "Do you want to add a rule to the security group to allow
 inbound traffic from your computer's IP address?"))
                {
                    await
 _autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, port,
 ipString);
                }
            }

            if (!sshPortIsOpen)
            {
                if (!interactive || GetYesNoResponse(
                        "Do you want to add a rule to the security group to allow
 inbound SSH traffic for debugging from your computer's IP address?"))
                {
```

```
                    await
_autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, sshPort,
ipString);
                }
            }
            loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);
        }

        if (loadBalancerAccess)
        {
            Console.WriteLine("Your load balancer is ready. You can access it by
browsing to:");
            Console.WriteLine($"\thttp://{endPoint}\n");
        }
        else
        {
            Console.WriteLine(
                "\nCouldn't get a successful response from the load balancer
endpoint. Troubleshoot by\n"
                + "manually verifying that your VPC and security group are
configured correctly and that\n"
                + "you can successfully make a GET request to the load balancer
endpoint:\n");
            Console.WriteLine($"\thttp://{endPoint}\n");
        }
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to continue with the
demo.");
        if (interactive)
            Console.ReadLine();
        return true;
    }

    /// <summary>
    /// Demonstrate the steps of the scenario.
    /// </summary>
    /// <param name="interactive">True to run as an interactive scenario.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> Demo(bool interactive)
    {
        var ssmOnlyPolicy = Path.Join(_configuration["resourcePath"],
            "ssm_only_policy.json");
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Resetting parameters to starting values for demo.");
        await _smParameterWrapper.Reset();

        Console.WriteLine("\nThis part of the demonstration shows how to toggle
different parts of the system\n" +
                          "to create situations where the web service fails, and
shows how using a resilient\n" +
                          "architecture can keep the web service running in spite
of these failures.");
        Console.WriteLine(new string('-', 88));
        Console.WriteLine("At the start, the load balancer endpoint returns
recommendations and reports that all targets are healthy.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine($"The web service running on the EC2 instances gets
recommendations by querying a DynamoDB table.\n" +
                          $"The table name is contained in a Systems Manager
parameter named '{_smParameterWrapper.TableParameter}'.\n" +
                          $"To simulate a failure of the recommendation service,
let's set this parameter to name a non-existent table.\n");
        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
"this-is-not-a-table");
        Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a failure code. But, the service reports as\n" +
                          "healthy to the load balancer because shallow health
checks don't check for failure of the recommendation service.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("Instead of failing when the recommendation service
fails, the web service can return a static response.");
        Console.WriteLine("While this is not a perfect solution, it presents the
customer with a somewhat better experience than failure.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.FailureResponseParameter,
"static");

        Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a static response.");
```

```
        Console.WriteLine("The service still reports as healthy because health
checks are still shallow.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("Let's reinstate the recommendation service.\n");
        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
_smParameterWrapper.TableName);
        Console.WriteLine(
            "\nLet's also substitute bad credentials for one of the instances in
the target group so that it can't\n" +
            "access the DynamoDB recommendation table.\n"
        );
        await _autoScalerWrapper.CreateInstanceProfileWithName(
            _autoScalerWrapper.BadCredsPolicyName,
            _autoScalerWrapper.BadCredsRoleName,
            _autoScalerWrapper.BadCredsProfileName,
            ssmOnlyPolicy,
            new List<string> { "AmazonSSMManagedInstanceCore" }
        );
        var instances = await
_autoScalerWrapper.GetInstancesByGroupName(_autoScalerWrapper.GroupName);
        var badInstanceId = instances.First();
        var instanceProfile = await
_autoScalerWrapper.GetInstanceProfile(badInstanceId);
        Console.WriteLine(
            $"Replacing the profile for instance {badInstanceId} with a profile
that contains\n" +
            "bad credentials...\n"
        );
        await _autoScalerWrapper.ReplaceInstanceProfile(
            badInstanceId,
            _autoScalerWrapper.BadCredsProfileName,
            instanceProfile.AssociationId
        );
        Console.WriteLine(
            "Now, sending a GET request to the load balancer endpoint returns
either a recommendation or a static response,\n" +
            "depending on which instance is selected by the load balancer.\n"
        );
        if (interactive)
            await DemoActionChoices();
```

```
        Console.WriteLine("\nLet's implement a deep health check. For this demo,
a deep health check tests whether");
        Console.WriteLine("the web service can access the DynamoDB table that it
depends on for recommendations. Note that");
        Console.WriteLine("the deep health check is only for ELB routing and not
for Auto Scaling instance health.");
        Console.WriteLine("This kind of deep health check is not recommended for
Auto Scaling instance health, because it");
        Console.WriteLine("risks accidental termination of all instances in the
Auto Scaling group when a dependent service fails.");

        Console.WriteLine("\nBy implementing deep health checks, the load
balancer can detect when one of the instances is failing");
        Console.WriteLine("and take that instance out of rotation.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.HealthCheckParameter,
"deep");

        Console.WriteLine($"\nNow, checking target health indicates that the
instance with bad credentials ({badInstanceId})");
        Console.WriteLine("is unhealthy. Note that it might take a minute or two
for the load balancer to detect the unhealthy");
        Console.WriteLine("instance. Sending a GET request to the load balancer
endpoint always returns a recommendation, because");
        Console.WriteLine("the load balancer takes unhealthy instances out of its
rotation.");

        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nBecause the instances in this demo are controlled by
an auto scaler, the simplest way to fix an unhealthy");
        Console.WriteLine("instance is to terminate it and let the auto scaler
start a new instance to replace it.");

        await _autoScalerWrapper.TryTerminateInstanceById(badInstanceId);

        Console.WriteLine($"\nEven while the instance is terminating and the new
instance is starting, sending a GET");
        Console.WriteLine("request to the web service continues to get a
successful recommendation response because");
        Console.WriteLine("starts and reports as healthy, it is included in the
load balancing rotation.");
```

```
        Console.WriteLine("Note that terminating and replacing an instance
typically takes several minutes, during which time you");
        Console.WriteLine("can see the changing health check status until the new
instance is running and healthy.");

        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nIf the recommendation service fails now, deep health
checks mean all instances report as unhealthy.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
"this-is-not-a-table");

        Console.WriteLine($"\nWhen all instances are unhealthy, the load balancer
continues to route requests even to");
        Console.WriteLine("unhealthy instances, allowing them to fail open and
return a static response rather than fail");
        Console.WriteLine("closed and report failure to the customer.");

        if (interactive)
            await DemoActionChoices();
        await _smParameterWrapper.Reset();

        Console.WriteLine(new string('-', 80));
        return true;
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="interactive">True to ask the user for cleanup.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> DestroyResources(bool interactive)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "To keep things tidy and to avoid unwanted charges on your account,
we can clean up all AWS resources\n" +
            "that were created for this demo."
        );
```

```
        if (!interactive || GetYesNoResponse("Do you want to clean up all demo
 resources? (y/n) "))
        {
            await
 _elasticLoadBalancerWrapper.DeleteLoadBalancerByName(_elasticLoadBalancerWrapper.LoadBal
            await
 _elasticLoadBalancerWrapper.DeleteTargetGroupByName(_elasticLoadBalancerWrapper.TargetGr
            await
 _autoScalerWrapper.TerminateAndDeleteAutoScalingGroupWithName(_autoScalerWrapper.GroupNa
            await
 _autoScalerWrapper.DeleteKeyPairByName(_autoScalerWrapper.KeyPairName);
            await
 _autoScalerWrapper.DeleteTemplateByName(_autoScalerWrapper.LaunchTemplateName);
            await _autoScalerWrapper.DeleteInstanceProfile(
                _autoScalerWrapper.BadCredsProfileName,
                _autoScalerWrapper.BadCredsRoleName
            );
            await
 _recommendations.DestroyDatabaseByName(_recommendations.TableName);
        }
        else
        {
            Console.WriteLine(
                "Ok, we'll leave the resources intact.\n" +
                "Don't forget to delete them when you're done with them or you
 might incur unexpected charges."
            );
        }

        Console.WriteLine(new string('-', 80));
        return true;
    }
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
/// <summary>
/// Encapsulates Amazon EC2 Auto Scaling and EC2 management methods.
/// </summary>
public class AutoScalerWrapper
{
    private readonly IAmazonAutoScaling _amazonAutoScaling;
    private readonly IAmazonEC2 _amazonEc2;
```

```
    private readonly IAmazonSimpleSystemsManagement _amazonSsm;
    private readonly IAmazonIdentityManagementService _amazonIam;

    private readonly string _instanceType = "";
    private readonly string _amiParam = "";
    private readonly string _launchTemplateName = "";
    private readonly string _groupName = "";
    private readonly string _instancePolicyName = "";
    private readonly string _instanceRoleName = "";
    private readonly string _instanceProfileName = "";
    private readonly string _badCredsProfileName = "";
    private readonly string _badCredsRoleName = "";
    private readonly string _badCredsPolicyName = "";
    private readonly string _keyPairName = "";

    public string GroupName => _groupName;
    public string KeyPairName => _keyPairName;
    public string LaunchTemplateName => _launchTemplateName;
    public string InstancePolicyName => _instancePolicyName;
    public string BadCredsProfileName => _badCredsProfileName;
    public string BadCredsRoleName => _badCredsRoleName;
    public string BadCredsPolicyName => _badCredsPolicyName;

    /// <summary>
    /// Constructor for the AutoScalerWrapper.
    /// </summary>
    /// <param name="amazonAutoScaling">The injected AutoScaling client.</param>
    /// <param name="amazonEc2">The injected EC2 client.</param>
    /// <param name="amazonIam">The injected IAM client.</param>
    /// <param name="amazonSsm">The injected SSM client.</param>
    public AutoScalerWrapper(
        IAmazonAutoScaling amazonAutoScaling,
        IAmazonEC2 amazonEc2,
        IAmazonSimpleSystemsManagement amazonSsm,
        IAmazonIdentityManagementService amazonIam,
        IConfiguration configuration)
    {
        _amazonAutoScaling = amazonAutoScaling;
        _amazonEc2 = amazonEc2;
        _amazonSsm = amazonSsm;
        _amazonIam = amazonIam;

        var prefix = configuration["resourcePrefix"];
        _instanceType = configuration["instanceType"];
```

```
        _amiParam = configuration["amiParam"];

        _launchTemplateName = prefix + "-template";
        _groupName = prefix + "-group";
        _instancePolicyName = prefix + "-pol";
        _instanceRoleName = prefix + "-role";
        _instanceProfileName = prefix + "-prof";
        _badCredsPolicyName = prefix + "-bc-pol";
        _badCredsRoleName = prefix + "-bc-role";
        _badCredsProfileName = prefix + "-bc-prof";
        _keyPairName = prefix + "-key-pair";
    }

    /// <summary>
    /// Create a policy, role, and profile that is associated with instances with
a specified name.
    /// An instance's associated profile defines a role that is assumed by the
    /// instance.The role has attached policies that specify the AWS permissions
granted to
    /// clients that run on the instance.
    /// </summary>
    /// <param name="policyName">Name to use for the policy.</param>
    /// <param name="roleName">Name to use for the role.</param>
    /// <param name="profileName">Name to use for the profile.</param>
    /// <param name="ssmOnlyPolicyFile">Path to a policy file for SSM.</param>
    /// <param name="awsManagedPolicies">AWS Managed policies to be attached to
the role.</param>
    /// <returns>The Arn of the profile.</returns>
    public async Task<string> CreateInstanceProfileWithName(
        string policyName,
        string roleName,
        string profileName,
        string ssmOnlyPolicyFile,
        List<string>? awsManagedPolicies = null)
    {

        var assumeRoleDoc = "{" +
                                    "\"Version\": \"2012-10-17\"," +
                                    "\"Statement\": [{" +
                                         "\"Effect\": \"Allow\"," +
                                         "\"Principal\": {" +
                                         "\"Service\": [" +
                                             "\"ec2.amazonaws.com\"" +
                                         "]" +
```

```
                                    "}," +
                                "\"Action\": \"sts:AssumeRole\"" +
                                "}]" +
                            "}";

        var policyDocument = await File.ReadAllTextAsync(ssmOnlyPolicyFile);

        var policyArn = "";

        try
        {
            var createPolicyResult = await _amazonIam.CreatePolicyAsync(
                new CreatePolicyRequest
                {
                    PolicyName = policyName,
                    PolicyDocument = policyDocument
                });
            policyArn = createPolicyResult.Policy.Arn;
        }
        catch (EntityAlreadyExistsException)
        {
            // The policy already exists, so we look it up to get the Arn.
            var policiesPaginator = _amazonIam.Paginators.ListPolicies(
                new ListPoliciesRequest()
                {
                    Scope = PolicyScopeType.Local
                });
            // Get the entire list using the paginator.
            await foreach (var policy in policiesPaginator.Policies)
            {
                if (policy.PolicyName.Equals(policyName))
                {
                    policyArn = policy.Arn;
                }
            }

            if (policyArn == null)
            {
                throw new InvalidOperationException("Policy not found");
            }
        }

        try
        {
```

```
            await _amazonIam.CreateRoleAsync(new CreateRoleRequest()
            {
                RoleName = roleName,
                AssumeRolePolicyDocument = assumeRoleDoc,
            });
            await _amazonIam.AttachRolePolicyAsync(new AttachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = policyArn
            });
            if (awsManagedPolicies != null)
            {
                foreach (var awsPolicy in awsManagedPolicies)
                {
                    await _amazonIam.AttachRolePolicyAsync(new
AttachRolePolicyRequest()
                    {
                        PolicyArn = $"arn:aws:iam::aws:policy/{awsPolicy}",
                        RoleName = roleName
                    });
                }
            }
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Role already exists.");
        }

        string profileArn = "";
        try
        {
            var profileCreateResponse = await
_amazonIam.CreateInstanceProfileAsync(
                new CreateInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            // Allow time for the profile to be ready.
            profileArn = profileCreateResponse.InstanceProfile.Arn;
            Thread.Sleep(10000);
            await _amazonIam.AddRoleToInstanceProfileAsync(
                new AddRoleToInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
```

```
                    RoleName = roleName
                });

        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine("Policy already exists.");
            var profileGetResponse = await _amazonIam.GetInstanceProfileAsync(
                new GetInstanceProfileRequest()
                {
                    InstanceProfileName = profileName
                });
            profileArn = profileGetResponse.InstanceProfile.Arn;
        }
        return profileArn;
    }

    /// <summary>
    /// Create a new key pair and save the file.
    /// </summary>
    /// <param name="newKeyPairName">The name of the new key pair.</param>
    /// <returns>Async task.</returns>
    public async Task CreateKeyPair(string newKeyPairName)
    {
        try
        {
            var keyResponse = await _amazonEc2.CreateKeyPairAsync(
                new CreateKeyPairRequest() { KeyName = newKeyPairName });
            await File.WriteAllTextAsync($"{newKeyPairName}.pem",
                keyResponse.KeyPair.KeyMaterial);
            Console.WriteLine($"Created key pair {newKeyPairName}.");
        }
        catch (AlreadyExistsException)
        {
            Console.WriteLine("Key pair already exists.");
        }
    }

    /// <summary>
    /// Delete the key pair and file by name.
    /// </summary>
    /// <param name="deleteKeyPairName">The key pair to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteKeyPairByName(string deleteKeyPairName)
```

```
    {
        try
        {
            await _amazonEc2.DeleteKeyPairAsync(
                new DeleteKeyPairRequest() { KeyName = deleteKeyPairName });
            File.Delete($"{deleteKeyPairName}.pem");
        }
        catch (FileNotFoundException)
        {
            Console.WriteLine($"Key pair {deleteKeyPairName} not found.");
        }
    }

    /// <summary>
    /// Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling.
    /// The launch template specifies a Bash script in its user data field that
runs after
    /// the instance is started. This script installs the Python packages and
starts a Python
    /// web server on the instance.
    /// </summary>
    /// <param name="startupScriptPath">The path to a Bash script file that is
run.</param>
    /// <param name="instancePolicyPath">The path to a permissions policy to
create and attach to the profile.</param>
    /// <returns>The template object.</returns>
    public async Task<Amazon.EC2.Model.LaunchTemplate> CreateTemplate(string
startupScriptPath, string instancePolicyPath)
    {
        await CreateKeyPair(_keyPairName);
        await CreateInstanceProfileWithName(_instancePolicyName,
_instanceRoleName, _instanceProfileName, instancePolicyPath);

        var startServerText = await File.ReadAllTextAsync(startupScriptPath);
        var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(startServerText);

        var amiLatest = await _amazonSsm.GetParameterAsync(
            new GetParameterRequest() { Name = _amiParam });
        var amiId = amiLatest.Parameter.Value;
        var launchTemplateResponse = await _amazonEc2.CreateLaunchTemplateAsync(
            new CreateLaunchTemplateRequest()
            {
                LaunchTemplateName = _launchTemplateName,
```

```
                LaunchTemplateData = new RequestLaunchTemplateData()
                {
                    InstanceType = _instanceType,
                    ImageId = amiId,
                    IamInstanceProfile =
                        new

 LaunchTemplateIamInstanceProfileSpecificationRequest()
                        {
                            Name = _instanceProfileName
                        },
                    KeyName = _keyPairName,
                    UserData = System.Convert.ToBase64String(plainTextBytes)
                }
            });
        return launchTemplateResponse.LaunchTemplate;

    }

    /// <summary>
    /// Get a list of Availability Zones in the AWS Region of the Amazon EC2
 Client.
    /// </summary>
    /// <returns>A list of availability zones.</returns>
    public async Task<List<string>> DescribeAvailabilityZones()
    {
        var zoneResponse = await _amazonEc2.DescribeAvailabilityZonesAsync(
            new DescribeAvailabilityZonesRequest());
        return zoneResponse.AvailabilityZones.Select(z => z.ZoneName).ToList();
    }

    /// <summary>
    /// Create an EC2 Auto Scaling group of a specified size and name.
    /// </summary>
    /// <param name="groupSize">The size for the group.</param>
    /// <param name="groupName">The name for the group.</param>
    /// <param name="availabilityZones">The availability zones for the group.</
param>
    /// <returns>Async task.</returns>
    public async Task CreateGroupOfSize(int groupSize, string groupName,
 List<string> availabilityZones)
    {
        try
        {
```

```
            await _amazonAutoScaling.CreateAutoScalingGroupAsync(
                new CreateAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName,
                    AvailabilityZones = availabilityZones,
                    LaunchTemplate =
                        new
Amazon.AutoScaling.Model.LaunchTemplateSpecification()
                        {
                            LaunchTemplateName = _launchTemplateName,
                            Version = "$Default"
                        },
                    MaxSize = groupSize,
                    MinSize = groupSize
                });
            Console.WriteLine($"Created EC2 Auto Scaling group {groupName} with
size {groupSize}.");
        }
        catch (EntityAlreadyExistsException)
        {
            Console.WriteLine($"EC2 Auto Scaling group {groupName} already
exists.");
        }
    }

    /// <summary>
    /// Get the default VPC for the account.
    /// </summary>
    /// <returns>The default VPC object.</returns>
    public async Task<Vpc> GetDefaultVpc()
    {
        var vpcResponse = await _amazonEc2.DescribeVpcsAsync(
            new DescribeVpcsRequest()
            {
                Filters = new List<Amazon.EC2.Model.Filter>()
                {
                    new ("is-default", new List<string>() { "true" })
                }
            });
        return vpcResponse.Vpcs[0];
    }

    /// <summary>
    /// Get all the subnets for a Vpc in a set of availability zones.
```

```
    /// </summary>
    /// <param name="vpcId">The Id of the Vpc.</param>
    /// <param name="availabilityZones">The list of availability zones.</param>
    /// <returns>The collection of subnet objects.</returns>
    public async Task<List<Subnet>> GetAllVpcSubnetsForZones(string vpcId,
List<string> availabilityZones)
    {
        var subnets = new List<Subnet>();
        var subnetPaginator = _amazonEc2.Paginators.DescribeSubnets(
            new DescribeSubnetsRequest()
            {
                Filters = new List<Amazon.EC2.Model.Filter>()
                {
                    new ("vpc-id", new List<string>() { vpcId}),
                    new ("availability-zone", availabilityZones),
                    new ("default-for-az", new List<string>() { "true" })
                }
            });

        // Get the entire list using the paginator.
        await foreach (var subnet in subnetPaginator.Subnets)
        {
            subnets.Add(subnet);
        }

        return subnets;
    }

    /// <summary>
    /// Delete a launch template by name.
    /// </summary>
    /// <param name="templateName">The name of the template to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteTemplateByName(string templateName)
    {
        try
        {
            await _amazonEc2.DeleteLaunchTemplateAsync(
                new DeleteLaunchTemplateRequest()
                {
                    LaunchTemplateName = templateName
                });
        }
        catch (AmazonClientException)
```

```
        {
            Console.WriteLine($"Unable to delete template {templateName}.");
        }
    }

    /// <summary>
    /// Detaches a role from an instance profile, detaches policies from the
role,
    /// and deletes all the resources.
    /// </summary>
    /// <param name="profileName">The name of the profile to delete.</param>
    /// <param name="roleName">The name of the role to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteInstanceProfile(string profileName, string roleName)
    {
        try
        {
            await _amazonIam.RemoveRoleFromInstanceProfileAsync(
                new RemoveRoleFromInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
            await _amazonIam.DeleteInstanceProfileAsync(
                new DeleteInstanceProfileRequest() { InstanceProfileName =
profileName });
            var attachedPolicies = await
_amazonIam.ListAttachedRolePoliciesAsync(
                new ListAttachedRolePoliciesRequest() { RoleName = roleName });
            foreach (var policy in attachedPolicies.AttachedPolicies)
            {
                await _amazonIam.DetachRolePolicyAsync(
                    new DetachRolePolicyRequest()
                    {
                        RoleName = roleName,
                        PolicyArn = policy.PolicyArn
                    });
                // Delete the custom policies only.
                if (!policy.PolicyArn.StartsWith("arn:aws:iam::aws"))
                {
                    await _amazonIam.DeletePolicyAsync(
                        new Amazon.IdentityManagement.Model.DeletePolicyRequest()
                        {
                            PolicyArn = policy.PolicyArn
```

```
                        });
                }
            }

            await _amazonIam.DeleteRoleAsync(
                new DeleteRoleRequest() { RoleName = roleName });
        }
        catch (NoSuchEntityException)
        {
            Console.WriteLine($"Instance profile {profileName} does not exist.");
        }
    }

    /// <summary>
    /// Gets data about the instances in an EC2 Auto Scaling group by its group
name.
    /// </summary>
    /// <param name="group">The name of the auto scaling group.</param>
    /// <returns>A collection of instance Ids.</returns>
    public async Task<IEnumerable<string>> GetInstancesByGroupName(string group)
    {
        var instanceResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
            new DescribeAutoScalingGroupsRequest()
            {
                AutoScalingGroupNames = new List<string>() { group }
            });
        var instanceIds = instanceResponse.AutoScalingGroups.SelectMany(
            g => g.Instances.Select(i => i.InstanceId));
        return instanceIds;
    }

    /// <summary>
    /// Get the instance profile association data for an instance.
    /// </summary>
    /// <param name="instanceId">The Id of the instance.</param>
    /// <returns>Instance profile associations data.</returns>
    public async Task<IamInstanceProfileAssociation> GetInstanceProfile(string
instanceId)
    {
        var response = await
_amazonEc2.DescribeIamInstanceProfileAssociationsAsync(
            new DescribeIamInstanceProfileAssociationsRequest()
            {
```

```
                Filters = new List<Amazon.EC2.Model.Filter>()
                {
                    new ("instance-id", new List<string>() { instanceId })
                },
            });
        return response.IamInstanceProfileAssociations[0];
    }

    /// <summary>
    /// Replace the profile associated with a running instance. After the profile
is replaced, the instance
    /// is rebooted to ensure that it uses the new profile. When the instance is
ready, Systems Manager is
    /// used to restart the Python web server.
    /// </summary>
    /// <param name="instanceId">The Id of the instance to update.</param>
    /// <param name="credsProfileName">The name of the new profile to associate
with the specified instance.</param>
    /// <param name="associationId">The Id of the existing profile association
for the instance.</param>
    /// <returns>Async task.</returns>
    public async Task ReplaceInstanceProfile(string instanceId, string
credsProfileName, string associationId)
    {
        await _amazonEc2.ReplaceIamInstanceProfileAssociationAsync(
                new ReplaceIamInstanceProfileAssociationRequest()
                {
                    AssociationId = associationId,
                    IamInstanceProfile = new IamInstanceProfileSpecification()
                    {
                        Name = credsProfileName
                    }
                });
        // Allow time before resetting.
        Thread.Sleep(25000);
        var instanceReady = false;
        var retries = 5;
        while (retries-- > 0 && !instanceReady)
        {
            await _amazonEc2.RebootInstancesAsync(
                new RebootInstancesRequest(new List<string>() { instanceId }));
            Thread.Sleep(10000);
```

```
            var instancesPaginator =
_amazonSsm.Paginators.DescribeInstanceInformation(
                new DescribeInstanceInformationRequest());
            // Get the entire list using the paginator.
            await foreach (var instance in
instancesPaginator.InstanceInformationList)
            {
                instanceReady = instance.InstanceId == instanceId;
                if (instanceReady)
                {
                    break;
                }
            }
        }
        Console.WriteLine($"Sending restart command to instance {instanceId}");
        await _amazonSsm.SendCommandAsync(
            new SendCommandRequest()
            {
                InstanceIds = new List<string>() { instanceId },
                DocumentName = "AWS-RunShellScript",
                Parameters = new Dictionary<string, List<string>>()
                {
                        {"commands", new List<string>() { "cd / && sudo python3
server.py 80" }}
                }
            });
        Console.WriteLine($"Restarted the web server on instance {instanceId}");
    }

    /// <summary>
    /// Try to terminate an instance by its Id.
    /// </summary>
    /// <param name="instanceId">The Id of the instance to terminate.</param>
    /// <returns>Async task.</returns>
    public async Task TryTerminateInstanceById(string instanceId)
    {
        var stopping = false;
        Console.WriteLine($"Stopping {instanceId}...");
        while (!stopping)
        {
            try
            {
                await
_amazonAutoScaling.TerminateInstanceInAutoScalingGroupAsync(
```

```
                    new TerminateInstanceInAutoScalingGroupRequest()
                    {
                        InstanceId = instanceId,
                        ShouldDecrementDesiredCapacity = false
                    });
                stopping = true;
            }
            catch (ScalingActivityInProgressException)
            {
                Console.WriteLine($"Scaling activity in progress for
{instanceId}. Waiting...");
                Thread.Sleep(10000);
            }
        }
    }

    /// <summary>
    /// Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
    /// waits and retries until the group is successfully deleted.
    /// </summary>
    /// <param name="groupName">The name of the group to try to delete.</param>
    /// <returns>Async task.</returns>
    public async Task TryDeleteGroupByName(string groupName)
    {
        var stopped = false;
        while (!stopped)
        {
            try
            {
                await _amazonAutoScaling.DeleteAutoScalingGroupAsync(
                    new DeleteAutoScalingGroupRequest()
                    {
                        AutoScalingGroupName = groupName
                    });
                stopped = true;
            }
            catch (Exception e)
                when ((e is ScalingActivityInProgressException)
                      || (e is Amazon.AutoScaling.Model.ResourceInUseException))
            {
                Console.WriteLine($"Some instances are still running.
Waiting...");
                Thread.Sleep(10000);
```

```
            }
        }
    }

    /// <summary>
    /// Terminate instances and delete the Auto Scaling group by name.
    /// </summary>
    /// <param name="groupName">The name of the group to delete.</param>
    /// <returns>Async task.</returns>
    public async Task TerminateAndDeleteAutoScalingGroupWithName(string
groupName)
    {
        var describeGroupsResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
            new DescribeAutoScalingGroupsRequest()
            {
                AutoScalingGroupNames = new List<string>() { groupName }
            });
        if (describeGroupsResponse.AutoScalingGroups.Any())
        {
            // Update the size to 0.
            await _amazonAutoScaling.UpdateAutoScalingGroupAsync(
                new UpdateAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName,
                    MinSize = 0
                });
            var group = describeGroupsResponse.AutoScalingGroups[0];
            foreach (var instance in group.Instances)
            {
                await TryTerminateInstanceById(instance.InstanceId);
            }

            await TryDeleteGroupByName(groupName);
        }
        else
        {
            Console.WriteLine($"No groups found with name {groupName}.");
        }
    }


    /// <summary>
    /// Get the default security group for a specified Vpc.
```

```
    /// </summary>
    /// <param name="vpc">The Vpc to search.</param>
    /// <returns>The default security group.</returns>
    public async Task<SecurityGroup> GetDefaultSecurityGroupForVpc(Vpc vpc)
    {
        var groupResponse = await _amazonEc2.DescribeSecurityGroupsAsync(
            new DescribeSecurityGroupsRequest()
            {
                Filters = new List<Amazon.EC2.Model.Filter>()
                {
                    new ("group-name", new List<string>() { "default" }),
                    new ("vpc-id", new List<string>() { vpc.VpcId })
                }
            });
        return groupResponse.SecurityGroups[0];
    }

    /// <summary>
    /// Verify the default security group of a Vpc allows ingress from the
calling computer.
    /// This can be done by allowing ingress from this computer's IP address.
    /// In some situations, such as connecting from a corporate network, you must
instead specify
    /// a prefix list Id. You can also temporarily open the port to any IP
address while running this example.
    /// If you do, be sure to remove public access when you're done.
    /// </summary>
    /// <param name="vpc">The group to check.</param>
    /// <param name="port">The port to verify.</param>
    /// <param name="ipAddress">This computer's IP address.</param>
    /// <returns>True if the ip address is allowed on the group.</returns>
    public bool VerifyInboundPortForGroup(SecurityGroup group, int port, string
ipAddress)
    {
        var portIsOpen = false;
        foreach (var ipPermission in group.IpPermissions)
        {
            if (ipPermission.FromPort == port)
            {
                foreach (var ipRange in ipPermission.Ipv4Ranges)
                {
                    var cidr = ipRange.CidrIp;
                    if (cidr.StartsWith(ipAddress) || cidr == "0.0.0.0/0")
                    {
```

```
                        portIsOpen = true;
                    }
                }

                if (ipPermission.PrefixListIds.Any())
                {
                    portIsOpen = true;
                }

                if (!portIsOpen)
                {
                    Console.WriteLine("The inbound rule does not appear to be
open to either this computer's IP\n" +
                                      "address, to all IP addresses (0.0.0.0/0),
or to a prefix list ID.");
                }
                else
                {
                    break;
                }
            }
        }

        return portIsOpen;
    }

    /// <summary>
    /// Add an ingress rule to the specified security group that allows access on
the
    /// specified port from the specified IP address.
    /// </summary>
    /// <param name="groupId">The Id of the security group to modify.</param>
    /// <param name="port">The port to open.</param>
    /// <param name="ipAddress">The IP address to allow access.</param>
    /// <returns>Async task.</returns>
    public async Task OpenInboundPort(string groupId, int port, string ipAddress)
    {
        await _amazonEc2.AuthorizeSecurityGroupIngressAsync(
            new AuthorizeSecurityGroupIngressRequest()
            {
                GroupId = groupId,
                IpPermissions = new List<IpPermission>()
                {
                    new IpPermission()
```

```
                    {
                        FromPort = port,
                        ToPort = port,
                        IpProtocol = "tcp",
                        Ipv4Ranges = new List<IpRange>()
                        {
                            new IpRange() { CidrIp = $"{ipAddress}/32" }
                        }
                    }
                }
            });
    }

    /// <summary>
    /// Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
 Scaling group.
    /// The
    /// </summary>
    /// <param name="autoScalingGroupName">The name of the Auto Scaling group.</
param>
    /// <param name="targetGroupArn">The Arn for the target group.</param>
    /// <returns>Async task.</returns>
    public async Task AttachLoadBalancerToGroup(string autoScalingGroupName,
 string targetGroupArn)
    {
        await _amazonAutoScaling.AttachLoadBalancerTargetGroupsAsync(
            new AttachLoadBalancerTargetGroupsRequest()
            {
                AutoScalingGroupName = autoScalingGroupName,
                TargetGroupARNs = new List<string>() { targetGroupArn }
            });
    }
}
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
/// <summary>
/// Encapsulates Elastic Load Balancer actions.
/// </summary>
public class ElasticLoadBalancerWrapper
{
```

```
    private readonly IAmazonElasticLoadBalancingV2 _amazonElasticLoadBalancingV2;
    private string? _endpoint = null;
    private readonly string _targetGroupName = "";
    private readonly string _loadBalancerName = "";
    HttpClient _httpClient = new();

    public string TargetGroupName => _targetGroupName;
    public string LoadBalancerName => _loadBalancerName;

    /// <summary>
    /// Constructor for the Elastic Load Balancer wrapper.
    /// </summary>
    /// <param name="amazonElasticLoadBalancingV2">The injected load balancing v2
client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public ElasticLoadBalancerWrapper(
        IAmazonElasticLoadBalancingV2 amazonElasticLoadBalancingV2,
        IConfiguration configuration)
    {
        _amazonElasticLoadBalancingV2 = amazonElasticLoadBalancingV2;
        var prefix = configuration["resourcePrefix"];
        _targetGroupName = prefix + "-tg";
        _loadBalancerName = prefix + "-lb";
    }

    /// <summary>
    /// Get the HTTP Endpoint of a load balancer by its name.
    /// </summary>
    /// <param name="loadBalancerName">The name of the load balancer.</param>
    /// <returns>The HTTP endpoint.</returns>
    public async Task<string> GetEndpointForLoadBalancerByName(string
loadBalancerName)
    {
        if (_endpoint == null)
        {
            var endpointResponse =
                await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                    new DescribeLoadBalancersRequest()
                    {
                        Names = new List<string>() { loadBalancerName }
                    });
            _endpoint = endpointResponse.LoadBalancers[0].DNSName;
        }
```

```
        return _endpoint;
    }

    /// <summary>
    /// Return the GET response for an endpoint as text.
    /// </summary>
    /// <param name="endpoint">The endpoint for the request.</param>
    /// <returns>The request response.</returns>
    public async Task<string> GetEndPointResponse(string endpoint)
    {
        var endpointResponse = await _httpClient.GetAsync($"http://{endpoint}");
        var textResponse = await endpointResponse.Content.ReadAsStringAsync();
        return textResponse!;
    }

    /// <summary>
    /// Get the target health for a group by name.
    /// </summary>
    /// <param name="groupName">The name of the group.</param>
    /// <returns>The collection of health descriptions.</returns>
    public async Task<List<TargetHealthDescription>>
CheckTargetHealthForGroup(string groupName)
    {
        List<TargetHealthDescription> result = null!;
        try
        {
            var groupResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                    new DescribeTargetGroupsRequest()
                    {
                        Names = new List<string>() { groupName }
                    });
            var healthResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetHealthAsync(
                    new DescribeTargetHealthRequest()
                    {
                        TargetGroupArn =
groupResponse.TargetGroups[0].TargetGroupArn
                    });
            ;
            result = healthResponse.TargetHealthDescriptions;
        }
        catch (TargetGroupNotFoundException)
        {
```

```
            Console.WriteLine($"Target group {groupName} not found.");
        }
        return result;
    }

    /// <summary>
    /// Create an Elastic Load Balancing target group. The target group specifies
 how the load balancer forwards
    /// requests to instances in the group and how instance health is checked.
    ///
    /// To speed up this demo, the health check is configured with shortened
 times and lower thresholds. In production,
    /// you might want to decrease the sensitivity of your health checks to avoid
 unwanted failures.
    /// </summary>
    /// <param name="groupName">The name for the group.</param>
    /// <param name="protocol">The protocol, such as HTTP.</param>
    /// <param name="port">The port to use to forward requests, such as 80.</
param>
    /// <param name="vpcId">The Id of the Vpc in which the load balancer
 exists.</param>
    /// <returns>The new TargetGroup object.</returns>
    public async Task<TargetGroup> CreateTargetGroupOnVpc(string groupName,
 ProtocolEnum protocol, int port, string vpcId)
    {
        var createResponse = await
 _amazonElasticLoadBalancingV2.CreateTargetGroupAsync(
            new CreateTargetGroupRequest()
            {
                Name = groupName,
                Protocol = protocol,
                Port = port,
                HealthCheckPath = "/healthcheck",
                HealthCheckIntervalSeconds = 10,
                HealthCheckTimeoutSeconds = 5,
                HealthyThresholdCount = 2,
                UnhealthyThresholdCount = 2,
                VpcId = vpcId
            });
        var targetGroup = createResponse.TargetGroups[0];
        return targetGroup;
    }

    /// <summary>
```

```
   /// Create an Elastic Load Balancing load balancer that uses the specified
subnets
   /// and forwards requests to the specified target group.
   /// </summary>
   /// <param name="name">The name for the new load balancer.</param>
   /// <param name="subnetIds">Subnets for the load balancer.</param>
   /// <param name="targetGroup">Target group for forwarded requests.</param>
   /// <returns>The new LoadBalancer object.</returns>
   public async Task<LoadBalancer> CreateLoadBalancerAndListener(string name,
List<string> subnetIds, TargetGroup targetGroup)
   {
       var createLbResponse = await
_amazonElasticLoadBalancingV2.CreateLoadBalancerAsync(
           new CreateLoadBalancerRequest()
           {
               Name = name,
               Subnets = subnetIds
           });
       var loadBalancerArn = createLbResponse.LoadBalancers[0].LoadBalancerArn;

       // Wait for load balancer to be available.
       var loadBalancerReady = false;
       while (!loadBalancerReady)
       {
           try
           {
               var describeResponse =
                   await
_amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                       new DescribeLoadBalancersRequest()
                       {
                           Names = new List<string>() { name }
                       });

               var loadBalancerState =
describeResponse.LoadBalancers[0].State.Code;

               loadBalancerReady = loadBalancerState ==
LoadBalancerStateEnum.Active;
           }
           catch (LoadBalancerNotFoundException)
           {
               loadBalancerReady = false;
           }
```

```
            Thread.Sleep(10000);
        }
        // Create the listener.
        await _amazonElasticLoadBalancingV2.CreateListenerAsync(
            new CreateListenerRequest()
            {
                LoadBalancerArn = loadBalancerArn,
                Protocol = targetGroup.Protocol,
                Port = targetGroup.Port,
                DefaultActions = new List<Action>()
                {
                    new Action()
                    {
                        Type = ActionTypeEnum.Forward,
                        TargetGroupArn = targetGroup.TargetGroupArn
                    }
                }
            });
        return createLbResponse.LoadBalancers[0];
    }

    /// <summary>
    /// Verify this computer can successfully send a GET request to the
    /// load balancer endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint to check.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> VerifyLoadBalancerEndpoint(string endpoint)
    {
        var success = false;
        var retries = 3;
        while (!success && retries > 0)
        {
            try
            {
                var endpointResponse = await _httpClient.GetAsync($"http://
{endpoint}");
                Console.WriteLine($"Response: {endpointResponse.StatusCode}.");

                if (endpointResponse.IsSuccessStatusCode)
                {
                    success = true;
                }
                else
```

```
                {
                    retries = 0;
                }
            }
            catch (HttpRequestException)
            {
                Console.WriteLine("Connection error, retrying...");
                retries--;
                Thread.Sleep(10000);
            }
        }

        return success;
    }

    /// <summary>
    /// Delete a load balancer by its specified name.
    /// </summary>
    /// <param name="name">The name of the load balancer to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteLoadBalancerByName(string name)
    {
        try
        {
            var describeLoadBalancerResponse =
                await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                    new DescribeLoadBalancersRequest()
                    {
                        Names = new List<string>() { name }
                    });
            var lbArn =
describeLoadBalancerResponse.LoadBalancers[0].LoadBalancerArn;
            await _amazonElasticLoadBalancingV2.DeleteLoadBalancerAsync(
                new DeleteLoadBalancerRequest()
                {
                    LoadBalancerArn = lbArn
                }
            );
        }
        catch (LoadBalancerNotFoundException)
        {
            Console.WriteLine($"Load balancer {name} not found.");
        }
    }
```

```
    /// <summary>
    /// Delete a TargetGroup by its specified name.
    /// </summary>
    /// <param name="groupName">Name of the group to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteTargetGroupByName(string groupName)
    {
        var done = false;
        while (!done)
        {
            try
            {
                var groupResponse =
                    await
_amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                        new DescribeTargetGroupsRequest()
                        {
                            Names = new List<string>() { groupName }
                        });

                var targetArn = groupResponse.TargetGroups[0].TargetGroupArn;
                await _amazonElasticLoadBalancingV2.DeleteTargetGroupAsync(
                    new DeleteTargetGroupRequest() { TargetGroupArn =
targetArn });
                Console.WriteLine($"Deleted load balancing target group
{groupName}.");
                done = true;
            }
            catch (TargetGroupNotFoundException)
            {
                Console.WriteLine(
                    $"Target group {groupName} not found, could not delete.");
                done = true;
            }
            catch (ResourceInUseException)
            {
                Console.WriteLine("Target group not yet released, waiting...");
                Thread.Sleep(10000);
            }
        }
    }
}
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
/// <summary>
/// Encapsulates a DynamoDB table to use as a service that recommends books,
 movies, and songs.
/// </summary>
public class Recommendations
{
    private readonly IAmazonDynamoDB _amazonDynamoDb;
    private readonly DynamoDBContext _context;
    private readonly string _tableName;

    public string TableName => _tableName;

    /// <summary>
    /// Constructor for the Recommendations service.
    /// </summary>
    /// <param name="amazonDynamoDb">The injected DynamoDb client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public Recommendations(IAmazonDynamoDB amazonDynamoDb, IConfiguration
 configuration)
    {
        _amazonDynamoDb = amazonDynamoDb;
        _context = new DynamoDBContext(_amazonDynamoDb);
        _tableName = configuration["databaseName"]!;
    }

    /// <summary>
    /// Create the DynamoDb table with a specified name.
    /// </summary>
    /// <param name="tableName">The name for the table.</param>
    /// <returns>True when ready.</returns>
    public async Task<bool> CreateDatabaseWithName(string tableName)
    {
        try
        {
            Console.Write($"Creating table {tableName}...");
            var createRequest = new CreateTableRequest()
            {
                TableName = tableName,
                AttributeDefinitions = new List<AttributeDefinition>()
```

```
                {
                    new AttributeDefinition()
                    {
                        AttributeName = "MediaType",
                        AttributeType = ScalarAttributeType.S
                    },
                    new AttributeDefinition()
                    {
                        AttributeName = "ItemId",
                        AttributeType = ScalarAttributeType.N
                    }
                },
            KeySchema = new List<KeySchemaElement>()
                {
                    new KeySchemaElement()
                    {
                        AttributeName = "MediaType",
                        KeyType = KeyType.HASH
                    },
                    new KeySchemaElement()
                    {
                        AttributeName = "ItemId",
                        KeyType = KeyType.RANGE
                    }
                },
            ProvisionedThroughput = new ProvisionedThroughput()
            {
                ReadCapacityUnits = 5,
                WriteCapacityUnits = 5
            }
        };
        await _amazonDynamoDb.CreateTableAsync(createRequest);

        // Wait until the table is ACTIVE and then report success.
        Console.Write("\nWaiting for table to become active...");

        var request = new DescribeTableRequest
        {
            TableName = tableName
        };

        TableStatus status;
        do
        {
```

```
                Thread.Sleep(2000);

                var describeTableResponse = await
 _amazonDynamoDb.DescribeTableAsync(request);
                status = describeTableResponse.Table.TableStatus;

                Console.Write(".");
            }
            while (status != "ACTIVE");

            return status == TableStatus.ACTIVE;
        }
        catch (ResourceInUseException)
        {
            Console.WriteLine($"Table {tableName} already exists.");
            return false;
        }
    }

    /// <summary>
    /// Populate the database table with data from a specified path.
    /// </summary>
    /// <param name="databaseTableName">The name of the table.</param>
    /// <param name="recommendationsPath">The path of the recommendations data.</
param>
    /// <returns>Async task.</returns>
    public async Task PopulateDatabase(string databaseTableName, string
 recommendationsPath)
    {
        var recommendationsText = await
 File.ReadAllTextAsync(recommendationsPath);
        var records =

 JsonSerializer.Deserialize<RecommendationModel[]>(recommendationsText);
        var batchWrite = _context.CreateBatchWrite<RecommendationModel>();

        foreach (var record in records!)
        {
            batchWrite.AddPutItem(record);
        }

        await batchWrite.ExecuteAsync();
    }
```

```
    /// <summary>
    /// Delete the recommendation table by name.
    /// </summary>
    /// <param name="tableName">The name of the recommendation table.</param>
    /// <returns>Async task.</returns>
    public async Task DestroyDatabaseByName(string tableName)
    {
        try
        {
            await _amazonDynamoDb.DeleteTableAsync(
                new DeleteTableRequest() { TableName = tableName });
            Console.WriteLine($"Table {tableName} was deleted.");
        }
        catch (ResourceNotFoundException)
        {
            Console.WriteLine($"Table {tableName} not found");
        }
    }
}
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
/// <summary>
/// Encapsulates Systems Manager parameter operations. This example uses these
 parameters
/// to drive the demonstration of resilient architecture, such as failure of a
 dependency or
/// how the service responds to a health check.
/// </summary>
public class SmParameterWrapper
{
    private readonly IAmazonSimpleSystemsManagement
 _amazonSimpleSystemsManagement;

    private readonly string _tableParameter = "doc-example-resilient-
architecture-table";
    private readonly string _failureResponseParameter = "doc-example-resilient-
architecture-failure-response";
    private readonly string _healthCheckParameter = "doc-example-resilient-
architecture-health-check";
    private readonly string _tableName = "";
```

```
    public string TableParameter => _tableParameter;
    public string TableName => _tableName;
    public string HealthCheckParameter => _healthCheckParameter;
    public string FailureResponseParameter => _failureResponseParameter;

    /// <summary>
    /// Constructor for the SmParameterWrapper.
    /// </summary>
    /// <param name="amazonSimpleSystemsManagement">The injected Simple Systems
 Management client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public SmParameterWrapper(IAmazonSimpleSystemsManagement
 amazonSimpleSystemsManagement, IConfiguration configuration)
    {
        _amazonSimpleSystemsManagement = amazonSimpleSystemsManagement;
        _tableName = configuration["databaseName"]!;
    }

    /// <summary>
    /// Reset the Systems Manager parameters to starting values for the demo.
    /// </summary>
    /// <returns>Async task.</returns>
    public async Task Reset()
    {
        await this.PutParameterByName(_tableParameter, _tableName);
        await this.PutParameterByName(_failureResponseParameter, "none");
        await this.PutParameterByName(_healthCheckParameter, "shallow");
    }

    /// <summary>
    /// Set the value of a named Systems Manager parameter.
    /// </summary>
    /// <param name="name">The name of the parameter.</param>
    /// <param name="value">The value to set.</param>
    /// <returns>Async task.</returns>
    public async Task PutParameterByName(string name, string value)
    {
        await _amazonSimpleSystemsManagement.PutParameterAsync(
            new PutParameterRequest() { Name = name, Value = value, Overwrite =
 true });
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - AttachLoadBalancerTargetGruppi
  - CreateAutoScalingGroup
  - CreateInstanceProfilo
  - CreateLaunchModello
  - CreateListener
  - CreateLoadBilanciatore
  - CreateTargetGruppo
  - DeleteAutoScalingGroup
  - DeleteInstanceProfilo
  - DeleteLaunchModello
  - DeleteLoadBilanciatore
  - DeleteTargetGruppo
  - DescribeAutoScalingGroups
  - DescribeAvailabilityZone
  - DescribeIamInstanceProfileAssociazioni
  - DescribeInstances
  - DescribeLoadBilanciatori
  - DescribeSubnets
  - DescribeTargetGruppi
  - DescribeTargetHealth
  - DescribeVpcs
  - RebootInstances
  - ReplaceIamInstanceProfileAssociazione
  - TerminateInstanceInAutoScalingGroup
  - UpdateAutoScalingGroup

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
public class Main {

    public static final String fileName = "C:\\AWS\\resworkflow\
\recommendations.json"; // Modify file location.
    public static final String tableName = "doc-example-recommendation-service";
    public static final String startScript = "C:\\AWS\\resworkflow\
\server_startup_script.sh"; // Modify file location.
    public static final String policyFile = "C:\\AWS\\resworkflow\
\instance_policy.json"; // Modify file location.
    public static final String ssmJSON = "C:\\AWS\\resworkflow\
\ssm_only_policy.json"; // Modify file location.
    public static final String failureResponse = "doc-example-resilient-
architecture-failure-response";
    public static final String healthCheck = "doc-example-resilient-architecture-
health-check";
    public static final String templateName = "doc-example-resilience-template";
    public static final String roleName = "doc-example-resilience-role";
    public static final String policyName = "doc-example-resilience-pol";
    public static final String profileName = "doc-example-resilience-prof";

    public static final String badCredsProfileName = "doc-example-resilience-
prof-bc";

    public static final String targetGroupName = "doc-example-resilience-tg";
    public static final String autoScalingGroupName = "doc-example-resilience-
group";
    public static final String lbName = "doc-example-resilience-lb";
    public static final String protocol = "HTTP";
    public static final int port = 80;
```

```
   public static final String DASHES = new String(new char[80]).replace("\0",
"-");

   public static void main(String[] args) throws IOException,
InterruptedException {
       Scanner in = new Scanner(System.in);
       Database database = new Database();
       AutoScaler autoScaler = new AutoScaler();
       LoadBalancer loadBalancer = new LoadBalancer();

       System.out.println(DASHES);
       System.out.println("Welcome to the demonstration of How to Build and
Manage a Resilient Service!");
       System.out.println(DASHES);

       System.out.println(DASHES);
       System.out.println("A - SETUP THE RESOURCES");
       System.out.println("Press Enter when you're ready to start deploying
resources.");
       in.nextLine();
       deploy(loadBalancer);
       System.out.println(DASHES);
       System.out.println(DASHES);
       System.out.println("B - DEMO THE RESILIENCE FUNCTIONALITY");
       System.out.println("Press Enter when you're ready.");
       in.nextLine();
       demo(loadBalancer);
       System.out.println(DASHES);

       System.out.println(DASHES);
       System.out.println("C - DELETE THE RESOURCES");
       System.out.println("""
               This concludes the demo of how to build and manage a resilient
service.
               To keep things tidy and to avoid unwanted charges on your
account, we can clean up all AWS resources
               that were created for this demo.
               """);

       System.out.println("\n Do you want to delete the resources (y/n)? ");
       String userInput = in.nextLine().trim().toLowerCase(); // Capture user
input

       if (userInput.equals("y")) {
```

```
            // Delete resources here
            deleteResources(loadBalancer, autoScaler, database);
            System.out.println("Resources deleted.");
        } else {
            System.out.println("""
                    Okay, we'll leave the resources intact.
                    Don't forget to delete them when you're done with them or you
might incur unexpected charges.
                    """);
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The example has completed. ");
        System.out.println("\n Thanks for watching!");
        System.out.println(DASHES);
    }

    // Deletes the AWS resources used in this example.
    private static void deleteResources(LoadBalancer loadBalancer, AutoScaler
autoScaler, Database database)
            throws IOException, InterruptedException {
        loadBalancer.deleteLoadBalancer(lbName);
        System.out.println("*** Wait 30 secs for resource to be deleted");
        TimeUnit.SECONDS.sleep(30);
        loadBalancer.deleteTargetGroup(targetGroupName);
        autoScaler.deleteAutoScaleGroup(autoScalingGroupName);
        autoScaler.deleteRolesPolicies(policyName, roleName, profileName);
        autoScaler.deleteTemplate(templateName);
        database.deleteTable(tableName);
    }

    private static void deploy(LoadBalancer loadBalancer) throws
InterruptedException, IOException {
        Scanner in = new Scanner(System.in);
        System.out.println(
                """
                        For this demo, we'll use the AWS SDK for Java (v2) to
create several AWS resources
                        to set up a load-balanced web service endpoint and
explore some ways to make it resilient
                        against various kinds of failures.

                        Some of the resources create by this demo are:
```

```
                        \t* A DynamoDB table that the web service depends on to
provide book, movie, and song recommendations.
                        \t* An EC2 launch template that defines EC2 instances
that each contain a Python web server.
                        \t* An EC2 Auto Scaling group that manages EC2 instances
across several Availability Zones.
                        \t* An Elastic Load Balancing (ELB) load balancer that
targets the Auto Scaling group to distribute requests.
                        """);

        System.out.println("Press Enter when you're ready.");
        in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Creating and populating a DynamoDB table named " +
tableName);
        Database database = new Database();
        database.createTable(tableName, fileName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
                Creating an EC2 launch template that runs '{startup_script}' when
an instance starts.
                This script starts a Python web server defined in the `server.py`
script. The web server
                listens to HTTP requests on port 80 and responds to requests to
'/' and to '/healthcheck'.
                For demo purposes, this server is run as the root user. In
production, the best practice is to
                run a web server, such as Apache, with least-privileged
credentials.

                The template also defines an IAM policy that each instance uses
to assume a role that grants
                permissions to access the DynamoDB recommendation table and
Systems Manager parameters
                that control the flow of the demo.
                """);

        LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
        templateCreator.createTemplate(policyFile, policyName, profileName,
startScript, templateName, roleName);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different Availability Zone.");
        System.out.println("*** Wait 30 secs for the VPC to be created");
        TimeUnit.SECONDS.sleep(30);
        AutoScaler autoScaler = new AutoScaler();
        String[] zones = autoScaler.createGroup(3, templateName,
autoScalingGroupName);

        System.out.println("""
                At this point, you have EC2 instances created. Once each instance
starts, it listens for
                HTTP requests. You can see these instances in the console or
continue with the demo.
                Press Enter when you're ready to continue.
                """);

        in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Creating variables that control the flow of the
demo.");
        ParameterHelper paramHelper = new ParameterHelper();
        paramHelper.reset();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
                Creating an Elastic Load Balancing target group and load
balancer. The target group
                defines how the load balancer connects to instances. The load
balancer provides a
                single endpoint where clients connect and dispatches requests to
instances in the group.
                """);

        String vpcId = autoScaler.getDefaultVPC();
        List<Subnet> subnets = autoScaler.getSubnets(vpcId, zones);
        System.out.println("You have retrieved a list with " + subnets.size() + "
subnets");
```

```
        String targetGroupArn = loadBalancer.createTargetGroup(protocol, port,
vpcId, targetGroupName);
        String elbDnsName = loadBalancer.createLoadBalancer(subnets,
targetGroupArn, lbName, port, protocol);
        autoScaler.attachLoadBalancerTargetGroup(autoScalingGroupName,
targetGroupArn);
        System.out.println("Verifying access to the load balancer endpoint...");
        boolean wasSuccessul =
loadBalancer.verifyLoadBalancerEndpoint(elbDnsName);
        if (!wasSuccessul) {
            System.out.println("Couldn't connect to the load balancer, verifying
that the port is open...");
            CloseableHttpClient httpClient = HttpClients.createDefault();

            // Create an HTTP GET request to "http://checkip.amazonaws.com"
            HttpGet httpGet = new HttpGet("http://checkip.amazonaws.com");
            try {
                // Execute the request and get the response
                HttpResponse response = httpClient.execute(httpGet);

                // Read the response content.
                String ipAddress =
IOUtils.toString(response.getEntity().getContent(),
StandardCharsets.UTF_8).trim();

                // Print the public IP address.
                System.out.println("Public IP Address: " + ipAddress);
                GroupInfo groupInfo = autoScaler.verifyInboundPort(vpcId, port,
ipAddress);
                if (!groupInfo.isPortOpen()) {
                    System.out.println("""
                            For this example to work, the default security group
for your default VPC must
                            allow access from this computer. You can either add
it automatically from this
                            example or add it yourself using the AWS Management
Console.
                            """);

                    System.out.println(
                            "Do you want to add a rule to security group " +
groupInfo.getGroupName() + " to allow");
                    System.out.println("inbound traffic on port " + port + " from
your computer's IP address (y/n) ");
```

```
                    String ans = in.nextLine();
                    if ("y".equalsIgnoreCase(ans)) {
                        autoScaler.openInboundPort(groupInfo.getGroupName(),
String.valueOf(port), ipAddress);
                        System.out.println("Security group rule added.");
                    } else {
                        System.out.println("No security group rule added.");
                    }
                }

            } catch (AutoScalingException e) {
                e.printStackTrace();
            }
        } else if (wasSuccessul) {
            System.out.println("Your load balancer is ready. You can access it by
browsing to:");
            System.out.println("\t http://" + elbDnsName);
        } else {
            System.out.println("Couldn't get a successful response from the load
balancer endpoint. Troubleshoot by");
            System.out.println("manually verifying that your VPC and security
group are configured correctly and that");
            System.out.println("you can successfully make a GET request to the
load balancer.");
        }

        System.out.println("Press Enter when you're ready to continue with the
demo.");
        in.nextLine();
    }

    // A method that controls the demo part of the Java program.
    public static void demo(LoadBalancer loadBalancer) throws IOException,
InterruptedException {
        ParameterHelper paramHelper = new ParameterHelper();
        System.out.println("Read the ssm_only_policy.json file");
        String ssmOnlyPolicy = readFileAsString(ssmJSON);

        System.out.println("Resetting parameters to starting values for demo.");
        paramHelper.reset();

        System.out.println(
                """
```

```
                        This part of the demonstration shows how to toggle
different parts of the system
                        to create situations where the web service fails, and
shows how using a resilient
                        architecture can keep the web service running in spite
of these failures.

                        At the start, the load balancer endpoint returns
recommendations and reports that all targets are healthy.
                       """);
       demoChoices(loadBalancer);

       System.out.println(
               """
                        The web service running on the EC2 instances gets
recommendations by querying a DynamoDB table.
                        The table name is contained in a Systems Manager
parameter named self.param_helper.table.
                        To simulate a failure of the recommendation service,
let's set this parameter to name a non-existent table.
                       """);
       paramHelper.put(paramHelper.tableName, "this-is-not-a-table");

       System.out.println(
               """
                        \nNow, sending a GET request to the load balancer
endpoint returns a failure code. But, the service reports as
                        healthy to the load balancer because shallow health
checks don't check for failure of the recommendation service.
                       """);
       demoChoices(loadBalancer);

       System.out.println(
               """
                       Instead of failing when the recommendation service fails,
the web service can return a static response.
                       While this is not a perfect solution, it presents the
customer with a somewhat better experience than failure.
                       """);
       paramHelper.put(paramHelper.failureResponse, "static");

       System.out.println("""
               Now, sending a GET request to the load balancer endpoint returns
a static response.
```

```
                The service still reports as healthy because health checks are
still shallow.
                """);
        demoChoices(loadBalancer);

        System.out.println("Let's reinstate the recommendation service.");
        paramHelper.put(paramHelper.tableName, paramHelper.dyntable);

        System.out.println("""
                Let's also substitute bad credentials for one of the instances in
the target group so that it can't
                access the DynamoDB recommendation table. We will get an instance
id value.
                """);

        LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
        AutoScaler autoScaler = new AutoScaler();

        // Create a new instance profile based on badCredsProfileName.
        templateCreator.createInstanceProfile(policyFile, policyName,
badCredsProfileName, roleName);
        String badInstanceId = autoScaler.getBadInstance(autoScalingGroupName);
        System.out.println("The bad instance id values used for this demo is " +
badInstanceId);

        String profileAssociationId =
autoScaler.getInstanceProfile(badInstanceId);
        System.out.println("The association Id value is " +
profileAssociationId);
        System.out.println("Replacing the profile for instance " + badInstanceId
                + " with a profile that contains bad credentials");
        autoScaler.replaceInstanceProfile(badInstanceId, badCredsProfileName,
profileAssociationId);

        System.out.println(
                """
                        Now, sending a GET request to the load balancer endpoint
returns either a recommendation or a static response,
                        depending on which instance is selected by the load
balancer.
                        """);

        demoChoices(loadBalancer);
```

```
        System.out.println("""
                Let's implement a deep health check. For this demo, a deep health
check tests whether
                the web service can access the DynamoDB table that it depends on
for recommendations. Note that
                the deep health check is only for ELB routing and not for Auto
Scaling instance health.
                This kind of deep health check is not recommended for Auto
Scaling instance health, because it
                risks accidental termination of all instances in the Auto Scaling
group when a dependent service fails.
                """);

        System.out.println("""
                By implementing deep health checks, the load balancer can detect
when one of the instances is failing
                and take that instance out of rotation.
                """);

        paramHelper.put(paramHelper.healthCheck, "deep");

        System.out.println("""
                Now, checking target health indicates that the instance with bad
credentials
                is unhealthy. Note that it might take a minute or two for the
load balancer to detect the unhealthy
                instance. Sending a GET request to the load balancer endpoint
always returns a recommendation, because
                the load balancer takes unhealthy instances out of its rotation.
                """);

        demoChoices(loadBalancer);

        System.out.println(
                """
                        Because the instances in this demo are controlled by an
auto scaler, the simplest way to fix an unhealthy
                        instance is to terminate it and let the auto scaler start
a new instance to replace it.
                        """);
        autoScaler.terminateInstance(badInstanceId);

        System.out.println("""
```

```
                Even while the instance is terminating and the new instance is
starting, sending a GET
                request to the web service continues to get a successful
recommendation response because
                the load balancer routes requests to the healthy instances. After
the replacement instance
                starts and reports as healthy, it is included in the load
balancing rotation.
                Note that terminating and replacing an instance typically takes
several minutes, during which time you
                can see the changing health check status until the new instance
is running and healthy.
                """);

        demoChoices(loadBalancer);
        System.out.println(
                "If the recommendation service fails now, deep health checks mean
all instances report as unhealthy.");
        paramHelper.put(paramHelper.tableName, "this-is-not-a-table");

        demoChoices(loadBalancer);
        paramHelper.reset();
    }

    public static void demoChoices(LoadBalancer loadBalancer) throws IOException,
InterruptedException {
        String[] actions = {
                "Send a GET request to the load balancer endpoint.",
                "Check the health of load balancer targets.",
                "Go to the next part of the demo."
        };
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("-".repeat(88));
            System.out.println("See the current state of the service by selecting
one of the following choices:");
            for (int i = 0; i < actions.length; i++) {
                System.out.println(i + ": " + actions[i]);
            }

            try {
                System.out.print("\nWhich action would you like to take? ");
                int choice = scanner.nextInt();
```

```
                System.out.println("-".repeat(88));

                switch (choice) {
                    case 0 -> {
                        System.out.println("Request:\n");
                        System.out.println("GET http://" +
loadBalancer.getEndpoint(lbName));
                        CloseableHttpClient httpClient =
HttpClients.createDefault();

                        // Create an HTTP GET request to the ELB.
                        HttpGet httpGet = new HttpGet("http://" +
loadBalancer.getEndpoint(lbName));

                        // Execute the request and get the response.
                        HttpResponse response = httpClient.execute(httpGet);
                        int statusCode =
response.getStatusLine().getStatusCode();
                        System.out.println("HTTP Status Code: " + statusCode);

                        // Display the JSON response
                        BufferedReader reader = new BufferedReader(
                                new
InputStreamReader(response.getEntity().getContent()));
                        StringBuilder jsonResponse = new StringBuilder();
                        String line;
                        while ((line = reader.readLine()) != null) {
                            jsonResponse.append(line);
                        }
                        reader.close();

                        // Print the formatted JSON response.
                        System.out.println("Full Response:\n");
                        System.out.println(jsonResponse.toString());

                        // Close the HTTP client.
                        httpClient.close();

                    }
                    case 1 -> {
                        System.out.println("\nChecking the health of load
balancer targets:\n");
                        List<TargetHealthDescription> health =
loadBalancer.checkTargetHealth(targetGroupName);
```

```
                        for (TargetHealthDescription target : health) {
                            System.out.printf("\tTarget %s on port %d is %s%n",
 target.target().id(),
                                    target.target().port(),
 target.targetHealth().stateAsString());
                        }
                        System.out.println("""
                                Note that it can take a minute or two for the
 health check to update
                                after changes are made.
                                """);
                    }
                    case 2 -> {
                        System.out.println("\nOkay, let's move on.");
                        System.out.println("-".repeat(88));
                        return; // Exit the method when choice is 2
                    }
                    default -> System.out.println("You must choose a value
 between 0-2. Please select again.");
                }

            } catch (java.util.InputMismatchException e) {
                System.out.println("Invalid input. Please select again.");
                scanner.nextLine(); // Clear the input buffer.
            }
        }
    }

    public static String readFileAsString(String filePath) throws IOException {
        byte[] bytes = Files.readAllBytes(Paths.get(filePath));
        return new String(bytes);
    }
}
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
public class AutoScaler {

    private static Ec2Client ec2Client;
    private static AutoScalingClient autoScalingClient;
    private static IamClient iamClient;
```

```
    private static SsmClient ssmClient;

    private IamClient getIAMClient() {
        if (iamClient == null) {
            iamClient = IamClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return iamClient;
    }

    private SsmClient getSSMClient() {
        if (ssmClient == null) {
            ssmClient = SsmClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return ssmClient;
    }

    private Ec2Client getEc2Client() {
        if (ec2Client == null) {
            ec2Client = Ec2Client.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return ec2Client;
    }

    private AutoScalingClient getAutoScalingClient() {
        if (autoScalingClient == null) {
            autoScalingClient = AutoScalingClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return autoScalingClient;
    }

    /**
     * Terminates and instances in an EC2 Auto Scaling group. After an instance
is
     * terminated, it can no longer be accessed.
     */
    public void terminateInstance(String instanceId) {
```

```
        TerminateInstanceInAutoScalingGroupRequest terminateInstanceIRequest =
TerminateInstanceInAutoScalingGroupRequest
                .builder()
                .instanceId(instanceId)
                .shouldDecrementDesiredCapacity(false)
                .build();


getAutoScalingClient().terminateInstanceInAutoScalingGroup(terminateInstanceIRequest);
        System.out.format("Terminated instance %s.", instanceId);
    }

    /**
     * Replaces the profile associated with a running instance. After the profile
is
     * replaced, the instance is rebooted to ensure that it uses the new profile.
     * When
     * the instance is ready, Systems Manager is used to restart the Python web
     * server.
     */
    public void replaceInstanceProfile(String instanceId, String
newInstanceProfileName, String profileAssociationId)
            throws InterruptedException {
        // Create an IAM instance profile specification.
        software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
iamInstanceProfile =
software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
                .builder()
                .name(newInstanceProfileName) // Make sure
'newInstanceProfileName' is a valid IAM Instance Profile
                                              // name.
                .build();

        // Replace the IAM instance profile association for the EC2 instance.
        ReplaceIamInstanceProfileAssociationRequest replaceRequest =
ReplaceIamInstanceProfileAssociationRequest
                .builder()
                .iamInstanceProfile(iamInstanceProfile)
                .associationId(profileAssociationId) // Make sure
'profileAssociationId' is a valid association ID.
                .build();

        try {
            getEc2Client().replaceIamInstanceProfileAssociation(replaceRequest);
```

```
            // Handle the response as needed.
        } catch (Ec2Exception e) {
            // Handle exceptions, log, or report the error.
            System.err.println("Error: " + e.getMessage());
        }
        System.out.format("Replaced instance profile for association %s with
profile %s.", profileAssociationId,
                newInstanceProfileName);
        TimeUnit.SECONDS.sleep(15);
        boolean instReady = false;
        int tries = 0;

        // Reboot after 60 seconds
        while (!instReady) {
            if (tries % 6 == 0) {
                getEc2Client().rebootInstances(RebootInstancesRequest.builder()
                        .instanceIds(instanceId)
                        .build());
                System.out.println("Rebooting instance " + instanceId + " and
waiting for it to be ready.");
            }
            tries++;
            try {
                TimeUnit.SECONDS.sleep(10);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            DescribeInstanceInformationResponse informationResponse =
getSSMClient().describeInstanceInformation();
            List<InstanceInformation> instanceInformationList =
informationResponse.instanceInformationList();
            for (InstanceInformation info : instanceInformationList) {
                if (info.instanceId().equals(instanceId)) {
                    instReady = true;
                    break;
                }
            }
        }

        SendCommandRequest sendCommandRequest = SendCommandRequest.builder()
                .instanceIds(instanceId)
                .documentName("AWS-RunShellScript")
                .parameters(Collections.singletonMap("commands",
```

```
                        Collections.singletonList("cd / && sudo python3 server.py
80")))
                .build();

        getSSMClient().sendCommand(sendCommandRequest);
        System.out.println("Restarted the Python web server on instance " +
instanceId + ".");
    }

    public void openInboundPort(String secGroupId, String port, String ipAddress)
{
        AuthorizeSecurityGroupIngressRequest ingressRequest =
AuthorizeSecurityGroupIngressRequest.builder()
                .groupName(secGroupId)
                .cidrIp(ipAddress)
                .fromPort(Integer.parseInt(port))
                .build();

        getEc2Client().authorizeSecurityGroupIngress(ingressRequest);
        System.out.format("Authorized ingress to %s on port %s from %s.",
secGroupId, port, ipAddress);
    }

    /**
     * Detaches a role from an instance profile, detaches policies from the role,
     * and deletes all the resources.
     */
    public void deleteInstanceProfile(String roleName, String profileName) {
        try {
            software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
getInstanceProfileRequest =
software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
                    .builder()
                    .instanceProfileName(profileName)
                    .build();

            GetInstanceProfileResponse response =
getIAMClient().getInstanceProfile(getInstanceProfileRequest);
            String name = response.instanceProfile().instanceProfileName();
            System.out.println(name);

            RemoveRoleFromInstanceProfileRequest profileRequest =
RemoveRoleFromInstanceProfileRequest.builder()
                    .instanceProfileName(profileName)
```

```
                    .roleName(roleName)
                    .build();

            getIAMClient().removeRoleFromInstanceProfile(profileRequest);
            DeleteInstanceProfileRequest deleteInstanceProfileRequest =
DeleteInstanceProfileRequest.builder()
                    .instanceProfileName(profileName)
                    .build();

            getIAMClient().deleteInstanceProfile(deleteInstanceProfileRequest);
            System.out.println("Deleted instance profile " + profileName);

            DeleteRoleRequest deleteRoleRequest = DeleteRoleRequest.builder()
                    .roleName(roleName)
                    .build();

            // List attached role policies.
            ListAttachedRolePoliciesResponse rolesResponse = getIAMClient()
                    .listAttachedRolePolicies(role -> role.roleName(roleName));
            List<AttachedPolicy> attachedPolicies =
rolesResponse.attachedPolicies();
            for (AttachedPolicy attachedPolicy : attachedPolicies) {
                DetachRolePolicyRequest request =
DetachRolePolicyRequest.builder()
                        .roleName(roleName)
                        .policyArn(attachedPolicy.policyArn())
                        .build();

                getIAMClient().detachRolePolicy(request);
                System.out.println("Detached and deleted policy " +
attachedPolicy.policyName());
            }

            getIAMClient().deleteRole(deleteRoleRequest);
            System.out.println("Instance profile and role deleted.");

        } catch (IamException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public void deleteTemplate(String templateName) {
```

```
        getEc2Client().deleteLaunchTemplate(name ->
name.launchTemplateName(templateName));
        System.out.format(templateName + " was deleted.");
    }

    public void deleteAutoScaleGroup(String groupName) {
        DeleteAutoScalingGroupRequest deleteAutoScalingGroupRequest =
DeleteAutoScalingGroupRequest.builder()
                .autoScalingGroupName(groupName)
                .forceDelete(true)
                .build();


getAutoScalingClient().deleteAutoScalingGroup(deleteAutoScalingGroupRequest);
        System.out.println(groupName + " was deleted.");
    }

    /*
     * Verify the default security group of the specified VPC allows ingress from
     * this
     * computer. This can be done by allowing ingress from this computer's IP
     * address. In some situations, such as connecting from a corporate network,
you
     * must instead specify a prefix list ID. You can also temporarily open the
port
     * to
     * any IP address while running this example. If you do, be sure to remove
     * public
     * access when you're done.
     *
     */
    public GroupInfo verifyInboundPort(String VPC, int port, String ipAddress) {
        boolean portIsOpen = false;
        GroupInfo groupInfo = new GroupInfo();
        try {
            Filter filter = Filter.builder()
                    .name("group-name")
                    .values("default")
                    .build();

            Filter filter1 = Filter.builder()
                    .name("vpc-id")
                    .values(VPC)
                    .build();
```

```
            DescribeSecurityGroupsRequest securityGroupsRequest =
DescribeSecurityGroupsRequest.builder()
                    .filters(filter, filter1)
                    .build();

            DescribeSecurityGroupsResponse securityGroupsResponse =
getEc2Client()
                    .describeSecurityGroups(securityGroupsRequest);
            String securityGroup =
securityGroupsResponse.securityGroups().get(0).groupName();
            groupInfo.setGroupName(securityGroup);

            for (SecurityGroup secGroup :
securityGroupsResponse.securityGroups()) {
                System.out.println("Found security group: " +
secGroup.groupId());

                for (IpPermission ipPermission : secGroup.ipPermissions()) {
                    if (ipPermission.fromPort() == port) {
                        System.out.println("Found inbound rule: " +
ipPermission);
                        for (IpRange ipRange : ipPermission.ipRanges()) {
                            String cidrIp = ipRange.cidrIp();
                            if (cidrIp.startsWith(ipAddress) ||
cidrIp.equals("0.0.0.0/0")) {
                                System.out.println(cidrIp + " is applicable");
                                portIsOpen = true;
                            }
                        }

                        if (!ipPermission.prefixListIds().isEmpty()) {
                            System.out.println("Prefix lList is applicable");
                            portIsOpen = true;
                        }

                        if (!portIsOpen) {
                            System.out
                                    .println("The inbound rule does not appear to
be open to either this computer's IP,"
                                            + " all IP addresses (0.0.0.0/0), or
to a prefix list ID.");
                        } else {
                            break;
```

```
                        }
                    }
                }
            }

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }

        groupInfo.setPortOpen(portIsOpen);
        return groupInfo;
    }

    /*
     * Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
     * Scaling group.
     * The target group specifies how the load balancer forward requests to the
     * instances
     * in the group.
     */
    public void attachLoadBalancerTargetGroup(String asGroupName, String
targetGroupARN) {
        try {
            AttachLoadBalancerTargetGroupsRequest targetGroupsRequest =
AttachLoadBalancerTargetGroupsRequest.builder()
                    .autoScalingGroupName(asGroupName)
                    .targetGroupARNs(targetGroupARN)
                    .build();


getAutoScalingClient().attachLoadBalancerTargetGroups(targetGroupsRequest);
            System.out.println("Attached load balancer to " + asGroupName);

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Creates an EC2 Auto Scaling group with the specified size.
    public String[] createGroup(int groupSize, String templateName, String
autoScalingGroupName) {

        // Get availability zones.
```

```
software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
zonesRequest =
software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
               .builder()
               .build();

       DescribeAvailabilityZonesResponse zonesResponse =
getEc2Client().describeAvailabilityZones(zonesRequest);
       List<String> availabilityZoneNames =
zonesResponse.availabilityZones().stream()

.map(software.amazon.awssdk.services.ec2.model.AvailabilityZone::zoneName)
               .collect(Collectors.toList());

       String availabilityZones = String.join(",", availabilityZoneNames);
       LaunchTemplateSpecification specification =
LaunchTemplateSpecification.builder()
               .launchTemplateName(templateName)
               .version("$Default")
               .build();

       String[] zones = availabilityZones.split(",");
       CreateAutoScalingGroupRequest groupRequest =
CreateAutoScalingGroupRequest.builder()
               .launchTemplate(specification)
               .availabilityZones(zones)
               .maxSize(groupSize)
               .minSize(groupSize)
               .autoScalingGroupName(autoScalingGroupName)
               .build();

       try {
           getAutoScalingClient().createAutoScalingGroup(groupRequest);

       } catch (AutoScalingException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       System.out.println("Created an EC2 Auto Scaling group named " +
autoScalingGroupName);
       return zones;
   }
```

```
   public String getDefaultVPC() {
       // Define the filter.
       Filter defaultFilter = Filter.builder()
               .name("is-default")
               .values("true")
               .build();

       software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest request =
software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest
               .builder()
               .filters(defaultFilter)
               .build();

       DescribeVpcsResponse response = getEc2Client().describeVpcs(request);
       return response.vpcs().get(0).vpcId();
   }

   // Gets the default subnets in a VPC for a specified list of Availability
Zones.
   public List<Subnet> getSubnets(String vpcId, String[] availabilityZones) {
       List<Subnet> subnets = null;
       Filter vpcFilter = Filter.builder()
               .name("vpc-id")
               .values(vpcId)
               .build();

       Filter azFilter = Filter.builder()
               .name("availability-zone")
               .values(availabilityZones)
               .build();

       Filter defaultForAZ = Filter.builder()
               .name("default-for-az")
               .values("true")
               .build();

       DescribeSubnetsRequest request = DescribeSubnetsRequest.builder()
               .filters(vpcFilter, azFilter, defaultForAZ)
               .build();

       DescribeSubnetsResponse response =
getEc2Client().describeSubnets(request);
       subnets = response.subnets();
       return subnets;
```

```
    }

    // Gets data about the instances in the EC2 Auto Scaling group.
    public String getBadInstance(String groupName) {
        DescribeAutoScalingGroupsRequest request =
DescribeAutoScalingGroupsRequest.builder()
                .autoScalingGroupNames(groupName)
                .build();

        DescribeAutoScalingGroupsResponse response =
getAutoScalingClient().describeAutoScalingGroups(request);
        AutoScalingGroup autoScalingGroup = response.autoScalingGroups().get(0);
        List<String> instanceIds = autoScalingGroup.instances().stream()
                .map(instance -> instance.instanceId())
                .collect(Collectors.toList());

        String[] instanceIdArray = instanceIds.toArray(new String[0]);
        for (String instanceId : instanceIdArray) {
            System.out.println("Instance ID: " + instanceId);
            return instanceId;
        }
        return "";
    }

    // Gets data about the profile associated with an instance.
    public String getInstanceProfile(String instanceId) {
        Filter filter = Filter.builder()
                .name("instance-id")
                .values(instanceId)
                .build();

        DescribeIamInstanceProfileAssociationsRequest associationsRequest =
DescribeIamInstanceProfileAssociationsRequest
                .builder()
                .filters(filter)
                .build();

        DescribeIamInstanceProfileAssociationsResponse response = getEc2Client()
                .describeIamInstanceProfileAssociations(associationsRequest);
        return response.iamInstanceProfileAssociations().get(0).associationId();
    }

    public void deleteRolesPolicies(String policyName, String roleName, String
InstanceProfile) {
```

```
        ListPoliciesRequest listPoliciesRequest =
 ListPoliciesRequest.builder().build();
        ListPoliciesResponse listPoliciesResponse =
 getIAMClient().listPolicies(listPoliciesRequest);
        for (Policy policy : listPoliciesResponse.policies()) {
            if (policy.policyName().equals(policyName)) {
                // List the entities (users, groups, roles) that are attached to
 the policy.

 software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
 listEntitiesRequest =
 software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
                        .builder()
                        .policyArn(policy.arn())
                        .build();
                ListEntitiesForPolicyResponse listEntitiesResponse = iamClient
                        .listEntitiesForPolicy(listEntitiesRequest);
                if (!listEntitiesResponse.policyGroups().isEmpty() || !
listEntitiesResponse.policyUsers().isEmpty()
                        || !listEntitiesResponse.policyRoles().isEmpty()) {
                    // Detach the policy from any entities it is attached to.
                    DetachRolePolicyRequest detachPolicyRequest =
 DetachRolePolicyRequest.builder()
                            .policyArn(policy.arn())
                            .roleName(roleName) // Specify the name of the IAM
 role
                            .build();

                    getIAMClient().detachRolePolicy(detachPolicyRequest);
                    System.out.println("Policy detached from entities.");
                }

                // Now, you can delete the policy.
                DeletePolicyRequest deletePolicyRequest =
 DeletePolicyRequest.builder()
                        .policyArn(policy.arn())
                        .build();

                getIAMClient().deletePolicy(deletePolicyRequest);
                System.out.println("Policy deleted successfully.");
                break;
            }
        }
```

```
        // List the roles associated with the instance profile
        ListInstanceProfilesForRoleRequest listRolesRequest =
 ListInstanceProfilesForRoleRequest.builder()
                .roleName(roleName)
                .build();

        // Detach the roles from the instance profile
        ListInstanceProfilesForRoleResponse listRolesResponse =
 iamClient.listInstanceProfilesForRole(listRolesRequest);
        for (software.amazon.awssdk.services.iam.model.InstanceProfile profile :
 listRolesResponse.instanceProfiles()) {
            RemoveRoleFromInstanceProfileRequest removeRoleRequest =
 RemoveRoleFromInstanceProfileRequest.builder()
                    .instanceProfileName(InstanceProfile)
                    .roleName(roleName) // Remove the extra dot here
                    .build();

            getIAMClient().removeRoleFromInstanceProfile(removeRoleRequest);
            System.out.println("Role " + roleName + " removed from instance
 profile " + InstanceProfile);
        }

        // Delete the instance profile after removing all roles
        DeleteInstanceProfileRequest deleteInstanceProfileRequest =
 DeleteInstanceProfileRequest.builder()
                .instanceProfileName(InstanceProfile)
                .build();

        getIAMClient().deleteInstanceProfile(r ->
 r.instanceProfileName(InstanceProfile));
        System.out.println(InstanceProfile + " Deleted");
        System.out.println("All roles and policies are deleted.");
    }
}
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
public class LoadBalancer {
    public ElasticLoadBalancingV2Client elasticLoadBalancingV2Client;

    public ElasticLoadBalancingV2Client getLoadBalancerClient() {
        if (elasticLoadBalancingV2Client == null) {
```

```
            elasticLoadBalancingV2Client = ElasticLoadBalancingV2Client.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }

        return elasticLoadBalancingV2Client;
    }

    // Checks the health of the instances in the target group.
    public List<TargetHealthDescription> checkTargetHealth(String
targetGroupName) {
        DescribeTargetGroupsRequest targetGroupsRequest =
DescribeTargetGroupsRequest.builder()
                .names(targetGroupName)
                .build();

        DescribeTargetGroupsResponse tgResponse =
getLoadBalancerClient().describeTargetGroups(targetGroupsRequest);

        DescribeTargetHealthRequest healthRequest =
DescribeTargetHealthRequest.builder()

.targetGroupArn(tgResponse.targetGroups().get(0).targetGroupArn())
                .build();

        DescribeTargetHealthResponse healthResponse =
getLoadBalancerClient().describeTargetHealth(healthRequest);
        return healthResponse.targetHealthDescriptions();
    }

    // Gets the HTTP endpoint of the load balancer.
    public String getEndpoint(String lbName) {
        DescribeLoadBalancersResponse res = getLoadBalancerClient()
                .describeLoadBalancers(describe -> describe.names(lbName));
        return res.loadBalancers().get(0).dnsName();
    }

    // Deletes a load balancer.
    public void deleteLoadBalancer(String lbName) {
        try {
            // Use a waiter to delete the Load Balancer.
            DescribeLoadBalancersResponse res = getLoadBalancerClient()
                    .describeLoadBalancers(describe -> describe.names(lbName));
```

```
            ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
            DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()

.loadBalancerArns(res.loadBalancers().get(0).loadBalancerArn())
                    .build();

            getLoadBalancerClient().deleteLoadBalancer(
                    builder ->
builder.loadBalancerArn(res.loadBalancers().get(0).loadBalancerArn()));
            WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
loadBalancerWaiter
                    .waitUntilLoadBalancersDeleted(request);
            waiterResponse.matched().response().ifPresent(System.out::println);

        } catch (ElasticLoadBalancingV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        System.out.println(lbName + " was deleted.");
    }

    // Deletes the target group.
    public void deleteTargetGroup(String targetGroupName) {
        try {
            DescribeTargetGroupsResponse res = getLoadBalancerClient()
                    .describeTargetGroups(describe ->
describe.names(targetGroupName));
            getLoadBalancerClient()
                    .deleteTargetGroup(builder ->
builder.targetGroupArn(res.targetGroups().get(0).targetGroupArn()));
        } catch (ElasticLoadBalancingV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        System.out.println(targetGroupName + " was deleted.");
    }

    // Verify this computer can successfully send a GET request to the load
balancer
    // endpoint.
    public boolean verifyLoadBalancerEndpoint(String elbDnsName) throws
IOException, InterruptedException {
        boolean success = false;
        int retries = 3;
```

```
        CloseableHttpClient httpClient = HttpClients.createDefault();

        // Create an HTTP GET request to the ELB.
        HttpGet httpGet = new HttpGet("http://" + elbDnsName);
        try {
            while ((!success) && (retries > 0)) {
                // Execute the request and get the response.
                HttpResponse response = httpClient.execute(httpGet);
                int statusCode = response.getStatusLine().getStatusCode();
                System.out.println("HTTP Status Code: " + statusCode);
                if (statusCode == 200) {
                    success = true;
                } else {
                    retries--;
                    System.out.println("Got connection error from load balancer
endpoint, retrying...");
                    TimeUnit.SECONDS.sleep(15);
                }
            }

        } catch (org.apache.http.conn.HttpHostConnectException e) {
            System.out.println(e.getMessage());
        }

        System.out.println("Status.." + success);
        return success;
    }

    /*
     * Creates an Elastic Load Balancing target group. The target group specifies
     * how
     * the load balancer forward requests to instances in the group and how
instance
     * health is checked.
     */
    public String createTargetGroup(String protocol, int port, String vpcId,
String targetGroupName) {
        CreateTargetGroupRequest targetGroupRequest =
CreateTargetGroupRequest.builder()
                .healthCheckPath("/healthcheck")
                .healthCheckTimeoutSeconds(5)
                .port(port)
                .vpcId(vpcId)
                .name(targetGroupName)
```

```
                .protocol(protocol)
                .build();

        CreateTargetGroupResponse targetGroupResponse =
getLoadBalancerClient().createTargetGroup(targetGroupRequest);
        String targetGroupArn =
targetGroupResponse.targetGroups().get(0).targetGroupArn();
        String targetGroup =
targetGroupResponse.targetGroups().get(0).targetGroupName();
        System.out.println("The " + targetGroup + " was created with ARN" +
targetGroupArn);
        return targetGroupArn;
    }

    /*
     * Creates an Elastic Load Balancing load balancer that uses the specified
     * subnets
     * and forwards requests to the specified target group.
     */
    public String createLoadBalancer(List<Subnet> subnetIds, String
targetGroupARN, String lbName, int port,
            String protocol) {
        try {
            List<String> subnetIdStrings = subnetIds.stream()
                    .map(Subnet::subnetId)
                    .collect(Collectors.toList());

            CreateLoadBalancerRequest balancerRequest =
CreateLoadBalancerRequest.builder()
                    .subnets(subnetIdStrings)
                    .name(lbName)
                    .scheme("internet-facing")
                    .build();

            // Create and wait for the load balancer to become available.
            CreateLoadBalancerResponse lsResponse =
getLoadBalancerClient().createLoadBalancer(balancerRequest);
            String lbARN = lsResponse.loadBalancers().get(0).loadBalancerArn();

            ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
            DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()
                    .loadBalancerArns(lbARN)
```

```
                    .build();

            System.out.println("Waiting for Load Balancer " + lbName + " to
 become available.");
            WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
 loadBalancerWaiter
                    .waitUntilLoadBalancerAvailable(request);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println("Load Balancer " + lbName + " is available.");

            // Get the DNS name (endpoint) of the load balancer.
            String lbDNSName = lsResponse.loadBalancers().get(0).dnsName();
            System.out.println("*** Load Balancer DNS Name: " + lbDNSName);

            // Create a listener for the load balance.
            Action action = Action.builder()
                    .targetGroupArn(targetGroupARN)
                    .type("forward")
                    .build();

            CreateListenerRequest listenerRequest =
 CreateListenerRequest.builder()

 .loadBalancerArn(lsResponse.loadBalancers().get(0).loadBalancerArn())
                    .defaultActions(action)
                    .port(port)
                    .protocol(protocol)
                    .defaultActions(action)
                    .build();

            getLoadBalancerClient().createListener(listenerRequest);
            System.out.println("Created listener to forward traffic from load
 balancer " + lbName + " to target group "
                    + targetGroupARN);

            // Return the load balancer DNS name.
            return lbDNSName;

        } catch (ElasticLoadBalancingV2Exception e) {
            e.printStackTrace();
        }
        return "";
    }
}
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
public class Database {

    private static DynamoDbClient dynamoDbClient;

    public static DynamoDbClient getDynamoDbClient() {
        if (dynamoDbClient == null) {
            dynamoDbClient = DynamoDbClient.builder()
                    .region(Region.US_EAST_1)
                    .build();
        }
        return dynamoDbClient;
    }

    // Checks to see if the Amazon DynamoDB table exists.
    private boolean doesTableExist(String tableName) {
        try {
            // Describe the table and catch any exceptions.
            DescribeTableRequest describeTableRequest =
 DescribeTableRequest.builder()
                    .tableName(tableName)
                    .build();

            getDynamoDbClient().describeTable(describeTableRequest);
            System.out.println("Table '" + tableName + "' exists.");
            return true;

        } catch (ResourceNotFoundException e) {
            System.out.println("Table '" + tableName + "' does not exist.");
        } catch (DynamoDbException e) {
            System.err.println("Error checking table existence: " +
 e.getMessage());
        }
        return false;
    }

    /*
     * Creates a DynamoDB table to use a recommendation service. The table has a
     * hash key named 'MediaType' that defines the type of media recommended,
 such
```

```
     * as
     * Book or Movie, and a range key named 'ItemId' that, combined with the
     * MediaType,
     * forms a unique identifier for the recommended item.
     */
    public void createTable(String tableName, String fileName) throws IOException
{
        // First check to see if the table exists.
        boolean doesExist = doesTableExist(tableName);
        if (!doesExist) {
            DynamoDbWaiter dbWaiter = getDynamoDbClient().waiter();
            CreateTableRequest createTableRequest = CreateTableRequest.builder()
                    .tableName(tableName)
                    .attributeDefinitions(
                            AttributeDefinition.builder()
                                    .attributeName("MediaType")
                                    .attributeType(ScalarAttributeType.S)
                                    .build(),
                            AttributeDefinition.builder()
                                    .attributeName("ItemId")
                                    .attributeType(ScalarAttributeType.N)
                                    .build())
                    .keySchema(
                            KeySchemaElement.builder()
                                    .attributeName("MediaType")
                                    .keyType(KeyType.HASH)
                                    .build(),
                            KeySchemaElement.builder()
                                    .attributeName("ItemId")
                                    .keyType(KeyType.RANGE)
                                    .build())
                    .provisionedThroughput(
                            ProvisionedThroughput.builder()
                                    .readCapacityUnits(5L)
                                    .writeCapacityUnits(5L)
                                    .build())
                    .build();

            getDynamoDbClient().createTable(createTableRequest);
            System.out.println("Creating table " + tableName + "...");

            // Wait until the Amazon DynamoDB table is created.
            DescribeTableRequest tableRequest = DescribeTableRequest.builder()
                    .tableName(tableName)
```

```
                    .build();

            WaiterResponse<DescribeTableResponse> waiterResponse =
dbWaiter.waitUntilTableExists(tableRequest);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println("Table " + tableName + " created.");

            // Add records to the table.
            populateTable(fileName, tableName);
        }
    }

    public void deleteTable(String tableName) {
        getDynamoDbClient().deleteTable(table -> table.tableName(tableName));
        System.out.println("Table " + tableName + " deleted.");
    }

    // Populates the table with data located in a JSON file using the DynamoDB
    // enhanced client.
    public void populateTable(String fileName, String tableName) throws
IOException {
        DynamoDbEnhancedClient enhancedClient = DynamoDbEnhancedClient.builder()
                .dynamoDbClient(getDynamoDbClient())
                .build();
        ObjectMapper objectMapper = new ObjectMapper();
        File jsonFile = new File(fileName);
        JsonNode rootNode = objectMapper.readTree(jsonFile);

        DynamoDbTable<Recommendation> mappedTable =
enhancedClient.table(tableName,
                TableSchema.fromBean(Recommendation.class));
        for (JsonNode currentNode : rootNode) {
            String mediaType = currentNode.path("MediaType").path("S").asText();
            int itemId = currentNode.path("ItemId").path("N").asInt();
            String title = currentNode.path("Title").path("S").asText();
            String creator = currentNode.path("Creator").path("S").asText();

            // Create a Recommendation object and set its properties.
            Recommendation rec = new Recommendation();
            rec.setMediaType(mediaType);
            rec.setItemId(itemId);
            rec.setTitle(title);
            rec.setCreator(creator);
```

```
            // Put the item into the DynamoDB table.
            mappedTable.putItem(rec); // Add the Recommendation to the list.
        }
        System.out.println("Added all records to the " + tableName);
    }
}
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
public class ParameterHelper {

    String tableName = "doc-example-resilient-architecture-table";
    String dyntable = "doc-example-recommendation-service";
    String failureResponse = "doc-example-resilient-architecture-failure-
response";
    String healthCheck = "doc-example-resilient-architecture-health-check";

    public void reset() {
        put(dyntable, tableName);
        put(failureResponse, "none");
        put(healthCheck, "shallow");
    }

    public void put(String name, String value) {
        SsmClient ssmClient = SsmClient.builder()
                .region(Region.US_EAST_1)
                .build();

        PutParameterRequest parameterRequest = PutParameterRequest.builder()
                .name(name)
                .value(value)
                .overwrite(true)
                .type("String")
                .build();

        ssmClient.putParameter(parameterRequest);
        System.out.printf("Setting demo parameter %s to '%s'.", name, value);
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - AttachLoadBalancerTargetGruppi
  - CreateAutoScalingGroup
  - CreateInstanceProfilo
  - CreateLaunchModello
  - CreateListener
  - CreateLoadBilanciatore
  - CreateTargetGruppo
  - DeleteAutoScalingGroup
  - DeleteInstanceProfilo
  - DeleteLaunchModello
  - DeleteLoadBilanciatore
  - DeleteTargetGruppo
  - DescribeAutoScalingGroups
  - DescribeAvailabilityZone
  - DescribeIamInstanceProfileAssociazioni
  - DescribeInstances
  - DescribeLoadBilanciatori
  - DescribeSubnets
  - DescribeTargetGruppi
  - DescribeTargetHealth
  - DescribeVpcs
  - RebootInstances
  - ReplaceIamInstanceProfileAssociazione
  - TerminateInstanceInAutoScalingGroup
  - UpdateAutoScalingGroup

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
#!/usr/bin/env node
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0


import {
  Scenario,
  parseScenarioArgs,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";

/**
 * The workflow steps are split into three stages:
 *   - deploy
 *   - demo
 *   - destroy
 *
 * Each of these stages has a corresponding file prefixed with steps-*.
 */
import { deploySteps } from "./steps-deploy.js";
import { demoSteps } from "./steps-demo.js";
import { destroySteps } from "./steps-destroy.js";

/**
 * The context is passed to every scenario. Scenario steps
 * will modify the context.
 */
const context = {};

/**
 * Three Scenarios are created for the workflow. A Scenario is an orchestration
 class
```

```
 * that simplifies running a series of steps.
 */
export const scenarios = {
  // Deploys all resources necessary for the workflow.
  deploy: new Scenario("Resilient Workflow - Deploy", deploySteps, context),
  // Demonstrates how a fragile web service can be made more resilient.
  demo: new Scenario("Resilient Workflow - Demo", demoSteps, context),
  // Destroys the resources created for the workflow.
  destroy: new Scenario("Resilient Workflow - Destroy", destroySteps, context),
};

// Call function if run directly
import { fileURLToPath } from "url";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  parseScenarioArgs(scenarios);
}
```

Crea passaggi per distribuire tutte le risorse.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { join } from "node:path";
import { readFileSync, writeFileSync } from "node:fs";
import axios from "axios";

import {
  BatchWriteItemCommand,
  CreateTableCommand,
  DynamoDBClient,
  waitUntilTableExists,
} from "@aws-sdk/client-dynamodb";
import {
  EC2Client,
  CreateKeyPairCommand,
  CreateLaunchTemplateCommand,
  DescribeAvailabilityZonesCommand,
  DescribeVpcsCommand,
  DescribeSubnetsCommand,
  DescribeSecurityGroupsCommand,
  AuthorizeSecurityGroupIngressCommand,
} from "@aws-sdk/client-ec2";
```

```
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  AttachRolePolicyCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import { SSMClient, GetParameterCommand } from "@aws-sdk/client-ssm";
import {
  CreateAutoScalingGroupCommand,
  AutoScalingClient,
  AttachLoadBalancerTargetGroupsCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  CreateListenerCommand,
  CreateLoadBalancerCommand,
  CreateTargetGroupCommand,
  ElasticLoadBalancingV2Client,
  waitUntilLoadBalancerAvailable,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
  ScenarioOutput,
  ScenarioInput,
  ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH, ROOT } from "./constants.js";
import { initParamsSteps } from "./steps-reset-params.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const deploySteps = [
  new ScenarioOutput("introduction", MESSAGES.introduction, { header: true }),
  new ScenarioInput("confirmDeployment", MESSAGES.confirmDeployment, {
    type: "confirm",
  }),
  new ScenarioAction(
    "handleConfirmDeployment",
    (c) => c.confirmDeployment === false && process.exit(),
```

```
  ),
  new ScenarioOutput(
    "creatingTable",
    MESSAGES.creatingTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioAction("createTable", async () => {
    const client = new DynamoDBClient({});
    await client.send(
      new CreateTableCommand({
        TableName: NAMES.tableName,
        ProvisionedThroughput: {
          ReadCapacityUnits: 5,
          WriteCapacityUnits: 5,
        },
        AttributeDefinitions: [
          {
            AttributeName: "MediaType",
            AttributeType: "S",
          },
          {
            AttributeName: "ItemId",
            AttributeType: "N",
          },
        ],
        KeySchema: [
          {
            AttributeName: "MediaType",
            KeyType: "HASH",
          },
          {
            AttributeName: "ItemId",
            KeyType: "RANGE",
          },
        ],
      }),
    );
    await waitUntilTableExists({ client }, { TableName: NAMES.tableName });
  }),
  new ScenarioOutput(
    "createdTable",
    MESSAGES.createdTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioOutput(
    "populatingTable",
```

```
    MESSAGES.populatingTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioAction("populateTable", () => {
    const client = new DynamoDBClient({});
    /**
     * @type {{ default: import("@aws-sdk/client-dynamodb").PutRequest['Item']
[] }}
     */
    const recommendations = JSON.parse(
      readFileSync(join(RESOURCES_PATH, "recommendations.json")),
    );

    return client.send(
      new BatchWriteItemCommand({
        RequestItems: {
          [NAMES.tableName]: recommendations.map((item) => ({
            PutRequest: { Item: item },
          })),
        },
      }),
    );
  }),
  new ScenarioOutput(
    "populatedTable",
    MESSAGES.populatedTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioOutput(
    "creatingKeyPair",
    MESSAGES.creatingKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
  ),
  new ScenarioAction("createKeyPair", async () => {
    const client = new EC2Client({});
    const { KeyMaterial } = await client.send(
      new CreateKeyPairCommand({
        KeyName: NAMES.keyPairName,
      }),
    );

    writeFileSync(`${NAMES.keyPairName}.pem`, KeyMaterial, { mode: 0o600 });
  }),
  new ScenarioOutput(
    "createdKeyPair",
    MESSAGES.createdKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
  ),
```

```
  new ScenarioOutput(
    "creatingInstancePolicy",
    MESSAGES.creatingInstancePolicy.replace(
      "${INSTANCE_POLICY_NAME}",
      NAMES.instancePolicyName,
    ),
  ),
  new ScenarioAction("createInstancePolicy", async (state) => {
    const client = new IAMClient({});
    const {
      Policy: { Arn },
    } = await client.send(
      new CreatePolicyCommand({
        PolicyName: NAMES.instancePolicyName,
        PolicyDocument: readFileSync(
          join(RESOURCES_PATH, "instance_policy.json"),
        ),
      }),
    );
    state.instancePolicyArn = Arn;
  }),
  new ScenarioOutput("createdInstancePolicy", (state) =>
    MESSAGES.createdInstancePolicy
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
      .replace("${INSTANCE_POLICY_ARN}", state.instancePolicyArn),
  ),
  new ScenarioOutput(
    "creatingInstanceRole",
    MESSAGES.creatingInstanceRole.replace(
      "${INSTANCE_ROLE_NAME}",
      NAMES.instanceRoleName,
    ),
  ),
  new ScenarioAction("createInstanceRole", () => {
    const client = new IAMClient({});
    return client.send(
      new CreateRoleCommand({
        RoleName: NAMES.instanceRoleName,
        AssumeRolePolicyDocument: readFileSync(
          join(ROOT, "assume-role-policy.json"),
        ),
      }),
    );
  }),
```

```
  new ScenarioOutput(
    "createdInstanceRole",
    MESSAGES.createdInstanceRole.replace(
      "${INSTANCE_ROLE_NAME}",
      NAMES.instanceRoleName,
    ),
  ),
  new ScenarioOutput(
    "attachingPolicyToRole",
    MESSAGES.attachingPolicyToRole
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName)
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName),
  ),
  new ScenarioAction("attachPolicyToRole", async (state) => {
    const client = new IAMClient({});
    await client.send(
      new AttachRolePolicyCommand({
        RoleName: NAMES.instanceRoleName,
        PolicyArn: state.instancePolicyArn,
      }),
    );
  }),
  new ScenarioOutput(
    "attachedPolicyToRole",
    MESSAGES.attachedPolicyToRole
      .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
  ),
  new ScenarioOutput(
    "creatingInstanceProfile",
    MESSAGES.creatingInstanceProfile.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.instanceProfileName,
    ),
  ),
  new ScenarioAction("createInstanceProfile", async (state) => {
    const client = new IAMClient({});
    const {
      InstanceProfile: { Arn },
    } = await client.send(
      new CreateInstanceProfileCommand({
        InstanceProfileName: NAMES.instanceProfileName,
      }),
    );
```

```
    state.instanceProfileArn = Arn;

    await waitUntilInstanceProfileExists(
      { client },
      { InstanceProfileName: NAMES.instanceProfileName },
    );
  }),
  new ScenarioOutput("createdInstanceProfile", (state) =>
    MESSAGES.createdInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_PROFILE_ARN}", state.instanceProfileArn),
  ),
  new ScenarioOutput(
    "addingRoleToInstanceProfile",
    MESSAGES.addingRoleToInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
  ),
  new ScenarioAction("addRoleToInstanceProfile", () => {
    const client = new IAMClient({});
    return client.send(
      new AddRoleToInstanceProfileCommand({
        RoleName: NAMES.instanceRoleName,
        InstanceProfileName: NAMES.instanceProfileName,
      }),
    );
  }),
  new ScenarioOutput(
    "addedRoleToInstanceProfile",
    MESSAGES.addedRoleToInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
  ),
  ...initParamsSteps,
  new ScenarioOutput("creatingLaunchTemplate", MESSAGES.creatingLaunchTemplate),
  new ScenarioAction("createLaunchTemplate", async () => {
    // snippet-start:[javascript.v3.wkflw.resilient.CreateLaunchTemplate]
    const ssmClient = new SSMClient({});
    const { Parameter } = await ssmClient.send(
      new GetParameterCommand({
        Name: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
      }),
    );
    const ec2Client = new EC2Client({});
```

```
    await ec2Client.send(
      new CreateLaunchTemplateCommand({
        LaunchTemplateName: NAMES.launchTemplateName,
        LaunchTemplateData: {
          InstanceType: "t3.micro",
          ImageId: Parameter.Value,
          IamInstanceProfile: { Name: NAMES.instanceProfileName },
          UserData: readFileSync(
            join(RESOURCES_PATH, "server_startup_script.sh"),
          ).toString("base64"),
          KeyName: NAMES.keyPairName,
        },
      }),
      // snippet-end:[javascript.v3.wkflw.resilient.CreateLaunchTemplate]
    );
  }),
  new ScenarioOutput(
    "createdLaunchTemplate",
    MESSAGES.createdLaunchTemplate.replace(
      "${LAUNCH_TEMPLATE_NAME}",
      NAMES.launchTemplateName,
    ),
  ),
  new ScenarioOutput(
    "creatingAutoScalingGroup",
    MESSAGES.creatingAutoScalingGroup.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    ),
  ),
  new ScenarioAction("createAutoScalingGroup", async (state) => {
    const ec2Client = new EC2Client({});
    const { AvailabilityZones } = await ec2Client.send(
      new DescribeAvailabilityZonesCommand({}),
    );
    state.availabilityZoneNames = AvailabilityZones.map((az) => az.ZoneName);
    const autoScalingClient = new AutoScalingClient({});
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new CreateAutoScalingGroupCommand({
          AvailabilityZones: state.availabilityZoneNames,
          AutoScalingGroupName: NAMES.autoScalingGroupName,
          LaunchTemplate: {
            LaunchTemplateName: NAMES.launchTemplateName,
```

```
          Version: "$Default",
        },
        MinSize: 3,
        MaxSize: 3,
      }),
    ),
  );
}),
new ScenarioOutput(
  "createdAutoScalingGroup",
  /**
   * @param {{ availabilityZoneNames: string[] }} state
   */
  (state) =>
    MESSAGES.createdAutoScalingGroup
      .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName)
      .replace(
        "${AVAILABILITY_ZONE_NAMES}",
        state.availabilityZoneNames.join(", "),
      ),
),
new ScenarioInput("confirmContinue", MESSAGES.confirmContinue, {
  type: "confirm",
}),
new ScenarioOutput("loadBalancer", MESSAGES.loadBalancer),
new ScenarioOutput("gettingVpc", MESSAGES.gettingVpc),
new ScenarioAction("getVpc", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeVpcs]
  const client = new EC2Client({});
  const { Vpcs } = await client.send(
    new DescribeVpcsCommand({
      Filters: [{ Name: "is-default", Values: ["true"] }],
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeVpcs]
  state.defaultVpc = Vpcs[0].VpcId;
}),
new ScenarioOutput("gotVpc", (state) =>
  MESSAGES.gotVpc.replace("${VPC_ID}", state.defaultVpc),
),
new ScenarioOutput("gettingSubnets", MESSAGES.gettingSubnets),
new ScenarioAction("getSubnets", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeSubnets]
  const client = new EC2Client({});
```

```
    const { Subnets } = await client.send(
      new DescribeSubnetsCommand({
        Filters: [
          { Name: "vpc-id", Values: [state.defaultVpc] },
          { Name: "availability-zone", Values: state.availabilityZoneNames },
          { Name: "default-for-az", Values: ["true"] },
        ],
      }),
    );
    // snippet-end:[javascript.v3.wkflw.resilient.DescribeSubnets]
    state.subnets = Subnets.map((subnet) => subnet.SubnetId);
  }),
  new ScenarioOutput(
    "gotSubnets",
    /**
     * @param {{ subnets: string[] }} state
     */
    (state) =>
      MESSAGES.gotSubnets.replace("${SUBNETS}", state.subnets.join(", ")),
  ),
  new ScenarioOutput(
    "creatingLoadBalancerTargetGroup",
    MESSAGES.creatingLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
      NAMES.loadBalancerTargetGroupName,
    ),
  ),
  new ScenarioAction("createLoadBalancerTargetGroup", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.CreateTargetGroup]
    const client = new ElasticLoadBalancingV2Client({});
    const { TargetGroups } = await client.send(
      new CreateTargetGroupCommand({
        Name: NAMES.loadBalancerTargetGroupName,
        Protocol: "HTTP",
        Port: 80,
        HealthCheckPath: "/healthcheck",
        HealthCheckIntervalSeconds: 10,
        HealthCheckTimeoutSeconds: 5,
        HealthyThresholdCount: 2,
        UnhealthyThresholdCount: 2,
        VpcId: state.defaultVpc,
      }),
    );
    // snippet-end:[javascript.v3.wkflw.resilient.CreateTargetGroup]
```

```
    const targetGroup = TargetGroups[0];
    state.targetGroupArn = targetGroup.TargetGroupArn;
    state.targetGroupProtocol = targetGroup.Protocol;
    state.targetGroupPort = targetGroup.Port;
  }),
  new ScenarioOutput(
    "createdLoadBalancerTargetGroup",
    MESSAGES.createdLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
      NAMES.loadBalancerTargetGroupName,
    ),
  ),
  new ScenarioOutput(
    "creatingLoadBalancer",
    MESSAGES.creatingLoadBalancer.replace("${LB_NAME}", NAMES.loadBalancerName),
  ),
  new ScenarioAction("createLoadBalancer", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.CreateLoadBalancer]
    const client = new ElasticLoadBalancingV2Client({});
    const { LoadBalancers } = await client.send(
      new CreateLoadBalancerCommand({
        Name: NAMES.loadBalancerName,
        Subnets: state.subnets,
      }),
    );
    state.loadBalancerDns = LoadBalancers[0].DNSName;
    state.loadBalancerArn = LoadBalancers[0].LoadBalancerArn;
    await waitUntilLoadBalancerAvailable(
      { client },
      { Names: [NAMES.loadBalancerName] },
    );
    // snippet-end:[javascript.v3.wkflw.resilient.CreateLoadBalancer]
  }),
  new ScenarioOutput("createdLoadBalancer", (state) =>
    MESSAGES.createdLoadBalancer
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${DNS_NAME}", state.loadBalancerDns),
  ),
  new ScenarioOutput(
    "creatingListener",
    MESSAGES.creatingLoadBalancerListener
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName),
  ),
```

```
  new ScenarioAction("createListener", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.CreateListener]
    const client = new ElasticLoadBalancingV2Client({});
    const { Listeners } = await client.send(
      new CreateListenerCommand({
        LoadBalancerArn: state.loadBalancerArn,
        Protocol: state.targetGroupProtocol,
        Port: state.targetGroupPort,
        DefaultActions: [
          { Type: "forward", TargetGroupArn: state.targetGroupArn },
        ],
      }),
    );
    // snippet-end:[javascript.v3.wkflw.resilient.CreateListener]
    const listener = Listeners[0];
    state.loadBalancerListenerArn = listener.ListenerArn;
  }),
  new ScenarioOutput("createdListener", (state) =>
    MESSAGES.createdLoadBalancerListener.replace(
      "${LB_LISTENER_ARN}",
      state.loadBalancerListenerArn,
    ),
  ),
  new ScenarioOutput(
    "attachingLoadBalancerTargetGroup",
    MESSAGES.attachingLoadBalancerTargetGroup
      .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName)
      .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName),
  ),
  new ScenarioAction("attachLoadBalancerTargetGroup", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.AttachTargetGroup]
    const client = new AutoScalingClient({});
    await client.send(
      new AttachLoadBalancerTargetGroupsCommand({
        AutoScalingGroupName: NAMES.autoScalingGroupName,
        TargetGroupARNs: [state.targetGroupArn],
      }),
    );
    // snippet-end:[javascript.v3.wkflw.resilient.AttachTargetGroup]
  }),
  new ScenarioOutput(
    "attachedLoadBalancerTargetGroup",
    MESSAGES.attachedLoadBalancerTargetGroup,
  ),
```

```
  new ScenarioOutput("verifyingInboundPort", MESSAGES.verifyingInboundPort),
  new ScenarioAction(
    "verifyInboundPort",
    /**
     *
     * @param {{ defaultSecurityGroup: import('@aws-sdk/client-
ec2').SecurityGroup}} state
     */
    async (state) => {
      const client = new EC2Client({});
      const { SecurityGroups } = await client.send(
        new DescribeSecurityGroupsCommand({
          Filters: [{ Name: "group-name", Values: ["default"] }],
        }),
      );
      if (!SecurityGroups) {
        state.verifyInboundPortError = new Error(MESSAGES.noSecurityGroups);
      }
      state.defaultSecurityGroup = SecurityGroups[0];

      /**
       * @type {string}
       */
      const ipResponse = (await axios.get("http://checkip.amazonaws.com")).data;
      state.myIp = ipResponse.trim();
      const myIpRules = state.defaultSecurityGroup.IpPermissions.filter(
        ({ IpRanges }) =>
          IpRanges.some(
            ({ CidrIp }) =>
              CidrIp.startsWith(state.myIp) || CidrIp === "0.0.0.0/0",
          ),
      )
        .filter(({ IpProtocol }) => IpProtocol === "tcp")
        .filter(({ FromPort }) => FromPort === 80);

      state.myIpRules = myIpRules;
    },
  ),
  new ScenarioOutput(
    "verifiedInboundPort",
    /**
     * @param {{ myIpRules: any[] }} state
     */
    (state) => {
```

```
      if (state.myIpRules.length > 0) {
        return MESSAGES.foundIpRules.replace(
          "${IP_RULES}",
          JSON.stringify(state.myIpRules, null, 2),
        );
      } else {
        return MESSAGES.noIpRules;
      }
    },
  ),
  new ScenarioInput(
    "shouldAddInboundRule",
    /**
     * @param {{ myIpRules: any[] }} state
     */
    (state) => {
      if (state.myIpRules.length > 0) {
        return false;
      } else {
        return MESSAGES.noIpRules;
      }
    },
    { type: "confirm" },
  ),
  new ScenarioAction(
    "addInboundRule",
    /**
     * @param {{ defaultSecurityGroup: import('@aws-sdk/client-
ec2').SecurityGroup }} state
     */
    async (state) => {
      if (!state.shouldAddInboundRule) {
        return;
      }

      const client = new EC2Client({});
      await client.send(
        new AuthorizeSecurityGroupIngressCommand({
          GroupId: state.defaultSecurityGroup.GroupId,
          CidrIp: `${state.myIp}/32`,
          FromPort: 80,
          ToPort: 80,
          IpProtocol: "tcp",
        }),
```

```
      );
    },
  ),
  new ScenarioOutput("addedInboundRule", (state) => {
    if (state.shouldAddInboundRule) {
      return MESSAGES.addedInboundRule.replace("${IP_ADDRESS}", state.myIp);
    } else {
      return false;
    }
  }),
  new ScenarioOutput("verifyingEndpoint", (state) =>
    MESSAGES.verifyingEndpoint.replace("${DNS_NAME}", state.loadBalancerDns),
  ),
  new ScenarioAction("verifyEndpoint", async (state) => {
    try {
      const response = await retry({ intervalInMs: 2000, maxRetries: 30 }, () =>
        axios.get(`http://${state.loadBalancerDns}`),
      );
      state.endpointResponse = JSON.stringify(response.data, null, 2);
    } catch (e) {
      state.verifyEndpointError = e;
    }
  }),
  new ScenarioOutput("verifiedEndpoint", (state) => {
    if (state.verifyEndpointError) {
      console.error(state.verifyEndpointError);
    } else {
      return MESSAGES.verifiedEndpoint.replace(
        "${ENDPOINT_RESPONSE}",
        state.endpointResponse,
      );
    }
  }),
];
```

Crea i passaggi per eseguire la demo.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { readFileSync } from "node:fs";
import { join } from "node:path";
```

```
import axios from "axios";

import {
  DescribeTargetGroupsCommand,
  DescribeTargetHealthCommand,
  ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";
import {
  DescribeInstanceInformationCommand,
  PutParameterCommand,
  SSMClient,
  SendCommandCommand,
} from "@aws-sdk/client-ssm";
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import {
  AutoScalingClient,
  DescribeAutoScalingGroupsCommand,
  TerminateInstanceInAutoScalingGroupCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  DescribeIamInstanceProfileAssociationsCommand,
  EC2Client,
  RebootInstancesCommand,
  ReplaceIamInstanceProfileAssociationCommand,
} from "@aws-sdk/client-ec2";

import {
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH } from "./constants.js";
import { findLoadBalancer } from "./shared.js";
```

```
const getRecommendation = new ScenarioAction(
  "getRecommendation",
  async (state) => {
    const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
    if (loadBalancer) {
      state.loadBalancerDnsName = loadBalancer.DNSName;
      try {
        state.recommendation = (
          await axios.get(`http://${state.loadBalancerDnsName}`)
        ).data;
      } catch (e) {
        state.recommendation = e instanceof Error ? e.message : e;
      }
    } else {
      throw new Error(MESSAGES.demoFindLoadBalancerError);
    }
  },
);

const getRecommendationResult = new ScenarioOutput(
  "getRecommendationResult",
  (state) =>
    `Recommendation:\n${JSON.stringify(state.recommendation, null, 2)}`,
  { preformatted: true },
);

const getHealthCheck = new ScenarioAction("getHealthCheck", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeTargetGroups]
  const client = new ElasticLoadBalancingV2Client({});
  const { TargetGroups } = await client.send(
    new DescribeTargetGroupsCommand({
      Names: [NAMES.loadBalancerTargetGroupName],
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeTargetGroups]

  // snippet-start:[javascript.v3.wkflw.resilient.DescribeTargetHealth]
  const { TargetHealthDescriptions } = await client.send(
    new DescribeTargetHealthCommand({
      TargetGroupArn: TargetGroups[0].TargetGroupArn,
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeTargetHealth]
  state.targetHealthDescriptions = TargetHealthDescriptions;
```

```
});

const getHealthCheckResult = new ScenarioOutput(
  "getHealthCheckResult",
  /**
   * @param {{ targetHealthDescriptions: import('@aws-sdk/client-elastic-load-
balancing-v2').TargetHealthDescription[]}} state
   */
  (state) => {
    const status = state.targetHealthDescriptions
      .map((th) => `${th.Target.Id}: ${th.TargetHealth.State}`)
      .join("\n");
    return `Health check:\n${status}`;
  },
  { preformatted: true },
);

const loadBalancerLoop = new ScenarioAction(
  "loadBalancerLoop",
  getRecommendation.action,
  {
    whileConfig: {
      whileFn: ({ loadBalancerCheck }) => loadBalancerCheck,
      input: new ScenarioInput(
        "loadBalancerCheck",
        MESSAGES.demoLoadBalancerCheck,
        {
          type: "confirm",
        },
      ),
      output: getRecommendationResult,
    },
  },
);

const healthCheckLoop = new ScenarioAction(
  "healthCheckLoop",
  getHealthCheck.action,
  {
    whileConfig: {
      whileFn: ({ healthCheck }) => healthCheck,
      input: new ScenarioInput("healthCheck", MESSAGES.demoHealthCheck, {
        type: "confirm",
      }),
```

```
      output: getHealthCheckResult,
    },
  },
);

const statusSteps = [
  getRecommendation,
  getRecommendationResult,
  getHealthCheck,
  getHealthCheckResult,
];

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const demoSteps = [
  new ScenarioOutput("header", MESSAGES.demoHeader, { header: true }),
  new ScenarioOutput("sanityCheck", MESSAGES.demoSanityCheck),
  ...statusSteps,
  new ScenarioInput(
    "brokenDependencyConfirmation",
    MESSAGES.demoBrokenDependencyConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("brokenDependency", async (state) => {
    if (!state.brokenDependencyConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      state.badTableName = `fake-table-${Date.now()}`;
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmTableNameKey,
          Value: state.badTableName,
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
  new ScenarioOutput("testBrokenDependency", (state) =>
    MESSAGES.demoTestBrokenDependency.replace(
      "${TABLE_NAME}",
      state.badTableName,
```

```
    ),
  ),
  ...statusSteps,
  new ScenarioInput(
    "staticResponseConfirmation",
    MESSAGES.demoStaticResponseConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("staticResponse", async (state) => {
    if (!state.staticResponseConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmFailureResponseKey,
          Value: "static",
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
  new ScenarioOutput("testStaticResponse", MESSAGES.demoTestStaticResponse),
  ...statusSteps,
  new ScenarioInput(
    "badCredentialsConfirmation",
    MESSAGES.demoBadCredentialsConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("badCredentialsExit", (state) => {
    if (!state.badCredentialsConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("fixDynamoDBName", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
```

```
    );
  }),
  new ScenarioAction(
    "badCredentials",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-auto-
scaling').Instance }} state
     */
    async (state) => {
      await createSsmOnlyInstanceProfile();
      const autoScalingClient = new AutoScalingClient({});
      const { AutoScalingGroups } = await autoScalingClient.send(
        new DescribeAutoScalingGroupsCommand({
          AutoScalingGroupNames: [NAMES.autoScalingGroupName],
        }),
      );
      state.targetInstance = AutoScalingGroups[0].Instances[0];
      // snippet-start:
[javascript.v3.wkflw.resilient.DescribeIamInstanceProfileAssociations]
      const ec2Client = new EC2Client({});
      const { IamInstanceProfileAssociations } = await ec2Client.send(
        new DescribeIamInstanceProfileAssociationsCommand({
          Filters: [
            { Name: "instance-id", Values: [state.targetInstance.InstanceId] },
          ],
        }),
      );
      // snippet-end:
[javascript.v3.wkflw.resilient.DescribeIamInstanceProfileAssociations]
      state.instanceProfileAssociationId =
        IamInstanceProfileAssociations[0].AssociationId;
      // snippet-start:
[javascript.v3.wkflw.resilient.ReplaceIamInstanceProfileAssociation]
      await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
        ec2Client.send(
          new ReplaceIamInstanceProfileAssociationCommand({
            AssociationId: state.instanceProfileAssociationId,
            IamInstanceProfile: { Name: NAMES.ssmOnlyInstanceProfileName },
          }),
        ),
      );
      // snippet-end:
[javascript.v3.wkflw.resilient.ReplaceIamInstanceProfileAssociation]
```

```
      await ec2Client.send(
        new RebootInstancesCommand({
          InstanceIds: [state.targetInstance.InstanceId],
        }),
      );

      const ssmClient = new SSMClient({});
      await retry({ intervalInMs: 20000, maxRetries: 15 }, async () => {
        const { InstanceInformationList } = await ssmClient.send(
          new DescribeInstanceInformationCommand({}),
        );

        const instance = InstanceInformationList.find(
          (info) => info.InstanceId === state.targetInstance.InstanceId,
        );

        if (!instance) {
          throw new Error("Instance not found.");
        }
      });

      await ssmClient.send(
        new SendCommandCommand({
          InstanceIds: [state.targetInstance.InstanceId],
          DocumentName: "AWS-RunShellScript",
          Parameters: { commands: ["cd / && sudo python3 server.py 80"] },
        }),
      );
    },
  ),
  new ScenarioOutput(
    "testBadCredentials",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation}} state
     */
    (state) =>
      MESSAGES.demoTestBadCredentials.replace(
        "${INSTANCE_ID}",
        state.targetInstance.InstanceId,
      ),
  ),
  loadBalancerLoop,
  new ScenarioInput(
```

```
    "deepHealthCheckConfirmation",
    MESSAGES.demoDeepHealthCheckConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("deepHealthCheckExit", (state) => {
    if (!state.deepHealthCheckConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("deepHealthCheck", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmHealthCheckKey,
        Value: "deep",
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testDeepHealthCheck", MESSAGES.demoTestDeepHealthCheck),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "killInstanceConfirmation",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation }} state
     */
    (state) =>
      MESSAGES.demoKillInstanceConfirmation.replace(
        "${INSTANCE_ID}",
        state.targetInstance.InstanceId,
      ),
    { type: "confirm" },
  ),
  new ScenarioAction("killInstanceExit", (state) => {
    if (!state.killInstanceConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction(
    "killInstance",
    /**
```

```
     * @param {{ targetInstance: import('@aws-sdk/client-
ssm').InstanceInformation }} state
     */
    async (state) => {
      const client = new AutoScalingClient({});
      await client.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: state.targetInstance.InstanceId,
          ShouldDecrementDesiredCapacity: false,
        }),
      );
    },
  ),
  new ScenarioOutput("testKillInstance", MESSAGES.demoTestKillInstance),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput("failOpenConfirmation", MESSAGES.demoFailOpenConfirmation, {
    type: "confirm",
  }),
  new ScenarioAction("failOpenExit", (state) => {
    if (!state.failOpenConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("failOpen", () => {
    const client = new SSMClient({});
    return client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: `fake-table-${Date.now()}`,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testFailOpen", MESSAGES.demoFailOpenTest),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "resetTableConfirmation",
    MESSAGES.demoResetTableConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("resetTableExit", (state) => {
```

```
    if (!state.resetTableConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("resetTable", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testResetTable", MESSAGES.demoTestResetTable),
  healthCheckLoop,
  loadBalancerLoop,
];

async function createSsmOnlyInstanceProfile() {
  const iamClient = new IAMClient({});
  const { Policy } = await iamClient.send(
    new CreatePolicyCommand({
      PolicyName: NAMES.ssmOnlyPolicyName,
      PolicyDocument: readFileSync(
        join(RESOURCES_PATH, "ssm_only_policy.json"),
      ),
    }),
  );
  await iamClient.send(
    new CreateRoleCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      AssumeRolePolicyDocument: JSON.stringify({
        Version: "2012-10-17",
        Statement: [
          {
            Effect: "Allow",
            Principal: { Service: "ec2.amazonaws.com" },
            Action: "sts:AssumeRole",
          },
        ],
      }),
    }),
```

```
  );
  await iamClient.send(
    new AttachRolePolicyCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      PolicyArn: Policy.Arn,
    }),
  );
  await iamClient.send(
    new AttachRolePolicyCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
    }),
  );
  // snippet-start:[javascript.v3.wkflw.resilient.CreateInstanceProfile]
  const { InstanceProfile } = await iamClient.send(
    new CreateInstanceProfileCommand({
      InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
    }),
  );
  await waitUntilInstanceProfileExists(
    { client: iamClient },
    { InstanceProfileName: NAMES.ssmOnlyInstanceProfileName },
  );
  // snippet-end:[javascript.v3.wkflw.resilient.CreateInstanceProfile]
  await iamClient.send(
    new AddRoleToInstanceProfileCommand({
      InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
      RoleName: NAMES.ssmOnlyRoleName,
    }),
  );

  return InstanceProfile;
}
```

Crea i passaggi per distruggere tutte le risorse.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { unlinkSync } from "node:fs";

import { DynamoDBClient, DeleteTableCommand } from "@aws-sdk/client-dynamodb";
import {
```

```
  EC2Client,
  DeleteKeyPairCommand,
  DeleteLaunchTemplateCommand,
} from "@aws-sdk/client-ec2";
import {
  IAMClient,
  DeleteInstanceProfileCommand,
  RemoveRoleFromInstanceProfileCommand,
  DeletePolicyCommand,
  DeleteRoleCommand,
  DetachRolePolicyCommand,
  paginateListPolicies,
} from "@aws-sdk/client-iam";
import {
  AutoScalingClient,
  DeleteAutoScalingGroupCommand,
  TerminateInstanceInAutoScalingGroupCommand,
  UpdateAutoScalingGroupCommand,
  paginateDescribeAutoScalingGroups,
} from "@aws-sdk/client-auto-scaling";
import {
  DeleteLoadBalancerCommand,
  DeleteTargetGroupCommand,
  DescribeTargetGroupsCommand,
  ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
  ScenarioOutput,
  ScenarioInput,
  ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES } from "./constants.js";
import { findLoadBalancer } from "./shared.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const destroySteps = [
  new ScenarioInput("destroy", MESSAGES.destroy, { type: "confirm" }),
  new ScenarioAction(
    "abort",
```

```
    (state) => state.destroy === false && process.exit(),
  ),
  new ScenarioAction("deleteTable", async (c) => {
    try {
      const client = new DynamoDBClient({});
      await client.send(new DeleteTableCommand({ TableName: NAMES.tableName }));
    } catch (e) {
      c.deleteTableError = e;
    }
  }),
  new ScenarioOutput("deleteTableResult", (state) => {
    if (state.deleteTableError) {
      console.error(state.deleteTableError);
      return MESSAGES.deleteTableError.replace(
        "${TABLE_NAME}",
        NAMES.tableName,
      );
    } else {
      return MESSAGES.deletedTable.replace("${TABLE_NAME}", NAMES.tableName);
    }
  }),
  new ScenarioAction("deleteKeyPair", async (state) => {
    try {
      const client = new EC2Client({});
      await client.send(
        new DeleteKeyPairCommand({ KeyName: NAMES.keyPairName }),
      );
      unlinkSync(`${NAMES.keyPairName}.pem`);
    } catch (e) {
      state.deleteKeyPairError = e;
    }
  }),
  new ScenarioOutput("deleteKeyPairResult", (state) => {
    if (state.deleteKeyPairError) {
      console.error(state.deleteKeyPairError);
      return MESSAGES.deleteKeyPairError.replace(
        "${KEY_PAIR_NAME}",
        NAMES.keyPairName,
      );
    } else {
      return MESSAGES.deletedKeyPair.replace(
        "${KEY_PAIR_NAME}",
        NAMES.keyPairName,
      );
```

```
    }
  }),
  new ScenarioAction("detachPolicyFromRole", async (state) => {
    try {
      const client = new IAMClient({});
      const policy = await findPolicy(NAMES.instancePolicyName);

      if (!policy) {
        state.detachPolicyFromRoleError = new Error(
          `Policy ${NAMES.instancePolicyName} not found.`,
        );
      } else {
        await client.send(
          new DetachRolePolicyCommand({
            RoleName: NAMES.instanceRoleName,
            PolicyArn: policy.Arn,
          }),
        );
      }
    } catch (e) {
      state.detachPolicyFromRoleError = e;
    }
  }),
  new ScenarioOutput("detachedPolicyFromRole", (state) => {
    if (state.detachPolicyFromRoleError) {
      console.error(state.detachPolicyFromRoleError);
      return MESSAGES.detachPolicyFromRoleError
        .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    } else {
      return MESSAGES.detachedPolicyFromRole
        .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    }
  }),
  new ScenarioAction("deleteInstancePolicy", async (state) => {
    const client = new IAMClient({});
    const policy = await findPolicy(NAMES.instancePolicyName);

    if (!policy) {
      state.deletePolicyError = new Error(
        `Policy ${NAMES.instancePolicyName} not found.`,
      );
    } else {
```

```
      return client.send(
        new DeletePolicyCommand({
          PolicyArn: policy.Arn,
        }),
      );
    }
  }),
  new ScenarioOutput("deletePolicyResult", (state) => {
    if (state.deletePolicyError) {
      console.error(state.deletePolicyError);
      return MESSAGES.deletePolicyError.replace(
        "${INSTANCE_POLICY_NAME}",
        NAMES.instancePolicyName,
      );
    } else {
      return MESSAGES.deletedPolicy.replace(
        "${INSTANCE_POLICY_NAME}",
        NAMES.instancePolicyName,
      );
    }
  }),
  new ScenarioAction("removeRoleFromInstanceProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new RemoveRoleFromInstanceProfileCommand({
          RoleName: NAMES.instanceRoleName,
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
    } catch (e) {
      state.removeRoleFromInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("removeRoleFromInstanceProfileResult", (state) => {
    if (state.removeRoleFromInstanceProfile) {
      console.error(state.removeRoleFromInstanceProfileError);
      return MESSAGES.removeRoleFromInstanceProfileError
        .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    } else {
      return MESSAGES.removedRoleFromInstanceProfile
        .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
```

```
    }
  }),
  new ScenarioAction("deleteInstanceRole", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new DeleteRoleCommand({
          RoleName: NAMES.instanceRoleName,
        }),
      );
    } catch (e) {
      state.deleteInstanceRoleError = e;
    }
  }),
  new ScenarioOutput("deleteInstanceRoleResult", (state) => {
    if (state.deleteInstanceRoleError) {
      console.error(state.deleteInstanceRoleError);
      return MESSAGES.deleteInstanceRoleError.replace(
        "${INSTANCE_ROLE_NAME}",
        NAMES.instanceRoleName,
      );
    } else {
      return MESSAGES.deletedInstanceRole.replace(
        "${INSTANCE_ROLE_NAME}",
        NAMES.instanceRoleName,
      );
    }
  }),
  new ScenarioAction("deleteInstanceProfile", async (state) => {
    try {
      // snippet-start:[javascript.v3.wkflw.resilient.DeleteInstanceProfile]
      const client = new IAMClient({});
      await client.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
      // snippet-end:[javascript.v3.wkflw.resilient.DeleteInstanceProfile]
    } catch (e) {
      state.deleteInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("deleteInstanceProfileResult", (state) => {
    if (state.deleteInstanceProfileError) {
```

```
      console.error(state.deleteInstanceProfileError);
      return MESSAGES.deleteInstanceProfileError.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.instanceProfileName,
      );
    } else {
      return MESSAGES.deletedInstanceProfile.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.instanceProfileName,
      );
    }
  }),
  new ScenarioAction("deleteAutoScalingGroup", async (state) => {
    try {
      await terminateGroupInstances(NAMES.autoScalingGroupName);
      await retry({ intervalInMs: 60000, maxRetries: 60 }, async () => {
        await deleteAutoScalingGroup(NAMES.autoScalingGroupName);
      });
    } catch (e) {
      state.deleteAutoScalingGroupError = e;
    }
  }),
  new ScenarioOutput("deleteAutoScalingGroupResult", (state) => {
    if (state.deleteAutoScalingGroupError) {
      console.error(state.deleteAutoScalingGroupError);
      return MESSAGES.deleteAutoScalingGroupError.replace(
        "${AUTO_SCALING_GROUP_NAME}",
        NAMES.autoScalingGroupName,
      );
    } else {
      return MESSAGES.deletedAutoScalingGroup.replace(
        "${AUTO_SCALING_GROUP_NAME}",
        NAMES.autoScalingGroupName,
      );
    }
  }),
  new ScenarioAction("deleteLaunchTemplate", async (state) => {
    const client = new EC2Client({});
    try {
      // snippet-start:[javascript.v3.wkflw.resilient.DeleteLaunchTemplate]
      await client.send(
        new DeleteLaunchTemplateCommand({
          LaunchTemplateName: NAMES.launchTemplateName,
        }),
```

```
      );
      // snippet-end:[javascript.v3.wkflw.resilient.DeleteLaunchTemplate]
    } catch (e) {
      state.deleteLaunchTemplateError = e;
    }
  }),
  new ScenarioOutput("deleteLaunchTemplateResult", (state) => {
    if (state.deleteLaunchTemplateError) {
      console.error(state.deleteLaunchTemplateError);
      return MESSAGES.deleteLaunchTemplateError.replace(
        "${LAUNCH_TEMPLATE_NAME}",
        NAMES.launchTemplateName,
      );
    } else {
      return MESSAGES.deletedLaunchTemplate.replace(
        "${LAUNCH_TEMPLATE_NAME}",
        NAMES.launchTemplateName,
      );
    }
  }),
  new ScenarioAction("deleteLoadBalancer", async (state) => {
    try {
      // snippet-start:[javascript.v3.wkflw.resilient.DeleteLoadBalancer]
      const client = new ElasticLoadBalancingV2Client({});
      const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
      await client.send(
        new DeleteLoadBalancerCommand({
          LoadBalancerArn: loadBalancer.LoadBalancerArn,
        }),
      );
      await retry({ intervalInMs: 1000, maxRetries: 60 }, async () => {
        const lb = await findLoadBalancer(NAMES.loadBalancerName);
        if (lb) {
          throw new Error("Load balancer still exists.");
        }
      });
      // snippet-end:[javascript.v3.wkflw.resilient.DeleteLoadBalancer]
    } catch (e) {
      state.deleteLoadBalancerError = e;
    }
  }),
  new ScenarioOutput("deleteLoadBalancerResult", (state) => {
    if (state.deleteLoadBalancerError) {
      console.error(state.deleteLoadBalancerError);
```

```
      return MESSAGES.deleteLoadBalancerError.replace(
        "${LB_NAME}",
        NAMES.loadBalancerName,
      );
    } else {
      return MESSAGES.deletedLoadBalancer.replace(
        "${LB_NAME}",
        NAMES.loadBalancerName,
      );
    }
  }),
  new ScenarioAction("deleteLoadBalancerTargetGroup", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.DeleteTargetGroup]
    const client = new ElasticLoadBalancingV2Client({});
    try {
      const { TargetGroups } = await client.send(
        new DescribeTargetGroupsCommand({
          Names: [NAMES.loadBalancerTargetGroupName],
        }),
      );

      await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
        client.send(
          new DeleteTargetGroupCommand({
            TargetGroupArn: TargetGroups[0].TargetGroupArn,
          }),
        ),
      );
    } catch (e) {
      state.deleteLoadBalancerTargetGroupError = e;
    }
    // snippet-end:[javascript.v3.wkflw.resilient.DeleteTargetGroup]
  }),
  new ScenarioOutput("deleteLoadBalancerTargetGroupResult", (state) => {
    if (state.deleteLoadBalancerTargetGroupError) {
      console.error(state.deleteLoadBalancerTargetGroupError);
      return MESSAGES.deleteLoadBalancerTargetGroupError.replace(
        "${TARGET_GROUP_NAME}",
        NAMES.loadBalancerTargetGroupName,
      );
    } else {
      return MESSAGES.deletedLoadBalancerTargetGroup.replace(
        "${TARGET_GROUP_NAME}",
        NAMES.loadBalancerTargetGroupName,
```

```
      );
    }
  }),
  new ScenarioAction("detachSsmOnlyRoleFromProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new RemoveRoleFromInstanceProfileCommand({
          InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
          RoleName: NAMES.ssmOnlyRoleName,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyRoleFromProfileError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyRoleFromProfileResult", (state) => {
    if (state.detachSsmOnlyRoleFromProfileError) {
      console.error(state.detachSsmOnlyRoleFromProfileError);
      return MESSAGES.detachSsmOnlyRoleFromProfileError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
    } else {
      return MESSAGES.detachedSsmOnlyRoleFromProfile
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
    }
  }),
  new ScenarioAction("detachSsmOnlyCustomRolePolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
      await iamClient.send(
        new DetachRolePolicyCommand({
          RoleName: NAMES.ssmOnlyRoleName,
          PolicyArn: ssmOnlyPolicy.Arn,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyCustomRolePolicyError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyCustomRolePolicyResult", (state) => {
    if (state.detachSsmOnlyCustomRolePolicyError) {
```

```
      console.error(state.detachSsmOnlyCustomRolePolicyError);
      return MESSAGES.detachSsmOnlyCustomRolePolicyError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
    } else {
      return MESSAGES.detachedSsmOnlyCustomRolePolicy
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
    }
  }),
  new ScenarioAction("detachSsmOnlyAWSRolePolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DetachRolePolicyCommand({
          RoleName: NAMES.ssmOnlyRoleName,
          PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
        }),
      );
    } catch (e) {
      state.detachSsmOnlyAWSRolePolicyError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyAWSRolePolicyResult", (state) => {
    if (state.detachSsmOnlyAWSRolePolicyError) {
      console.error(state.detachSsmOnlyAWSRolePolicyError);
      return MESSAGES.detachSsmOnlyAWSRolePolicyError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
    } else {
      return MESSAGES.detachedSsmOnlyAWSRolePolicy
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
    }
  }),
  new ScenarioAction("deleteSsmOnlyInstanceProfile", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
        }),
      );
    } catch (e) {
```

```
      state.deleteSsmOnlyInstanceProfileError = e;
    }
  }),
  new ScenarioOutput("deleteSsmOnlyInstanceProfileResult", (state) => {
    if (state.deleteSsmOnlyInstanceProfileError) {
      console.error(state.deleteSsmOnlyInstanceProfileError);
      return MESSAGES.deleteSsmOnlyInstanceProfileError.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.ssmOnlyInstanceProfileName,
      );
    } else {
      return MESSAGES.deletedSsmOnlyInstanceProfile.replace(
        "${INSTANCE_PROFILE_NAME}",
        NAMES.ssmOnlyInstanceProfileName,
      );
    }
  }),
  new ScenarioAction("deleteSsmOnlyPolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
      await iamClient.send(
        new DeletePolicyCommand({
          PolicyArn: ssmOnlyPolicy.Arn,
        }),
      );
    } catch (e) {
      state.deleteSsmOnlyPolicyError = e;
    }
  }),
  new ScenarioOutput("deleteSsmOnlyPolicyResult", (state) => {
    if (state.deleteSsmOnlyPolicyError) {
      console.error(state.deleteSsmOnlyPolicyError);
      return MESSAGES.deleteSsmOnlyPolicyError.replace(
        "${POLICY_NAME}",
        NAMES.ssmOnlyPolicyName,
      );
    } else {
      return MESSAGES.deletedSsmOnlyPolicy.replace(
        "${POLICY_NAME}",
        NAMES.ssmOnlyPolicyName,
      );
    }
  }),
```

```
  new ScenarioAction("deleteSsmOnlyRole", async (state) => {
    try {
      const iamClient = new IAMClient({});
      await iamClient.send(
        new DeleteRoleCommand({
          RoleName: NAMES.ssmOnlyRoleName,
        }),
      );
    } catch (e) {
      state.deleteSsmOnlyRoleError = e;
    }
  }),
  new ScenarioOutput("deleteSsmOnlyRoleResult", (state) => {
    if (state.deleteSsmOnlyRoleError) {
      console.error(state.deleteSsmOnlyRoleError);
      return MESSAGES.deleteSsmOnlyRoleError.replace(
        "${ROLE_NAME}",
        NAMES.ssmOnlyRoleName,
      );
    } else {
      return MESSAGES.deletedSsmOnlyRole.replace(
        "${ROLE_NAME}",
        NAMES.ssmOnlyRoleName,
      );
    }
  }),
];

/**
 * @param {string} policyName
 */
async function findPolicy(policyName) {
  const client = new IAMClient({});
  const paginatedPolicies = paginateListPolicies({ client }, {});
  for await (const page of paginatedPolicies) {
    const policy = page.Policies.find((p) => p.PolicyName === policyName);
    if (policy) {
      return policy;
    }
  }
}

/**
 * @param {string} groupName
```

```
 */
async function deleteAutoScalingGroup(groupName) {
  const client = new AutoScalingClient({});
  try {
    await client.send(
      new DeleteAutoScalingGroupCommand({
        AutoScalingGroupName: groupName,
      }),
    );
  } catch (err) {
    if (!(err instanceof Error)) {
      throw err;
    } else {
      console.log(err.name);
      throw err;
    }
  }
}

/**
 * @param {string} groupName
 */
async function terminateGroupInstances(groupName) {
  const autoScalingClient = new AutoScalingClient({});
  const group = await findAutoScalingGroup(groupName);
  await autoScalingClient.send(
    new UpdateAutoScalingGroupCommand({
      AutoScalingGroupName: group.AutoScalingGroupName,
      MinSize: 0,
    }),
  );
  for (const i of group.Instances) {
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: i.InstanceId,
          ShouldDecrementDesiredCapacity: true,
        }),
      ),
    );
  }
}

async function findAutoScalingGroup(groupName) {
```

```
  const client = new AutoScalingClient({});
  const paginatedGroups = paginateDescribeAutoScalingGroups({ client }, {});
  for await (const page of paginatedGroups) {
    const group = page.AutoScalingGroups.find(
      (g) => g.AutoScalingGroupName === groupName,
    );
    if (group) {
      return group;
    }
  }
  throw new Error(`Auto scaling group ${groupName} not found.`);
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - AttachLoadBalancerTargetGruppi
  - CreateAutoScalingGroup
  - CreateInstanceProfilo
  - CreateLaunchModello
  - CreateListener
  - CreateLoadBilanciatore
  - CreateTargetGruppo
  - DeleteAutoScalingGroup
  - DeleteInstanceProfilo
  - DeleteLaunchModello
  - DeleteLoadBilanciatore
  - DeleteTargetGruppo
  - DescribeAutoScalingGroups
  - DescribeAvailabilityZone
  - DescribeIamInstanceProfileAssociazioni
  - DescribeInstances
  - DescribeLoadBilanciatori
  - DescribeSubnets
  - DescribeTargetGruppi

- DescribeTargetHealth
- DescribeVpcs
- RebootInstances
- ReplaceIamInstanceProfileAssociazione
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui lo scenario interattivo al prompt dei comandi.

```
class Runner:
    def __init__(
        self, resource_path, recommendation, autoscaler, loadbalancer,
 param_helper
    ):
        self.resource_path = resource_path
        self.recommendation = recommendation
        self.autoscaler = autoscaler
        self.loadbalancer = loadbalancer
        self.param_helper = param_helper
        self.protocol = "HTTP"
        self.port = 80
        self.ssh_port = 22

    def deploy(self):
        recommendations_path = f"{self.resource_path}/recommendations.json"
        startup_script = f"{self.resource_path}/server_startup_script.sh"
        instance_policy = f"{self.resource_path}/instance_policy.json"

        print(
```

```
            "\nFor this demo, we'll use the AWS SDK for Python (Boto3) to create
several AWS resources\n"
            "to set up a load-balanced web service endpoint and explore some ways
to make it resilient\n"
            "against various kinds of failures.\n\n"
            "Some of the resources create by this demo are:\n"
        )
        print(
            "\t* A DynamoDB table that the web service depends on to provide
book, movie, and song recommendations."
        )
        print(
            "\t* An EC2 launch template that defines EC2 instances that each
contain a Python web server."
        )
        print(
            "\t* An EC2 Auto Scaling group that manages EC2 instances across
several Availability Zones."
        )
        print(
            "\t* An Elastic Load Balancing (ELB) load balancer that targets the
Auto Scaling group to distribute requests."
        )
        print("-" * 88)
        q.ask("Press Enter when you're ready to start deploying resources.")

        print(
            f"Creating and populating a DynamoDB table named
'{self.recommendation.table_name}'."
        )
        self.recommendation.create()
        self.recommendation.populate(recommendations_path)
        print("-" * 88)

        print(
            f"Creating an EC2 launch template that runs '{startup_script}' when
an instance starts.\n"
            f"This script starts a Python web server defined in the `server.py`
script. The web server\n"
            f"listens to HTTP requests on port 80 and responds to requests to '/'
and to '/healthcheck'.\n"
            f"For demo purposes, this server is run as the root user. In
production, the best practice is to\n"
```

```
            f"run a web server, such as Apache, with least-privileged
credentials.\n"
        )
        print(
            f"The template also defines an IAM policy that each instance uses to
assume a role that grants\n"
            f"permissions to access the DynamoDB recommendation table and Systems
Manager parameters\n"
            f"that control the flow of the demo.\n"
        )
        self.autoscaler.create_template(startup_script, instance_policy)
        print("-" * 88)

        print(
            f"Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different\n"
            f"Availability Zone."
        )
        zones = self.autoscaler.create_group(3)
        print("-" * 88)
        print(
            "At this point, you have EC2 instances created. Once each instance
starts, it listens for\n"
            "HTTP requests. You can see these instances in the console or
continue with the demo."
        )
        print("-" * 88)
        q.ask("Press Enter when you're ready to continue.")

        print(f"Creating variables that control the flow of the demo.\n")
        self.param_helper.reset()

        print(
            "\nCreating an Elastic Load Balancing target group and load balancer.
The target group\n"
            "defines how the load balancer connects to instances. The load
balancer provides a\n"
            "single endpoint where clients connect and dispatches requests to
instances in the group.\n"
        )
        vpc = self.autoscaler.get_default_vpc()
        subnets = self.autoscaler.get_subnets(vpc["VpcId"], zones)
        target_group = self.loadbalancer.create_target_group(
            self.protocol, self.port, vpc["VpcId"]
```

```
        )
        self.loadbalancer.create_load_balancer(
            [subnet["SubnetId"] for subnet in subnets], target_group
        )
        self.autoscaler.attach_load_balancer_target_group(target_group)
        print(f"Verifying access to the load balancer endpoint...")
        lb_success = self.loadbalancer.verify_load_balancer_endpoint()
        if not lb_success:
            print(
                "Couldn't connect to the load balancer, verifying that the port
 is open..."
            )
            current_ip_address = requests.get(
                "http://checkip.amazonaws.com"
            ).text.strip()
            sec_group, port_is_open = self.autoscaler.verify_inbound_port(
                vpc, self.port, current_ip_address
            )
            sec_group, ssh_port_is_open = self.autoscaler.verify_inbound_port(
                vpc, self.ssh_port, current_ip_address
            )
            if not port_is_open:
                print(
                    "For this example to work, the default security group for
 your default VPC must\n"
                    "allows access from this computer. You can either add it
 automatically from this\n"
                    "example or add it yourself using the AWS Management Console.
\n"
                )
                if q.ask(
                    f"Do you want to add a rule to security group
 {sec_group['GroupId']} to allow\n"
                    f"inbound traffic on port {self.port} from your computer's IP
 address of {current_ip_address}? (y/n) ",
                    q.is_yesno,
                ):
                    self.autoscaler.open_inbound_port(
                        sec_group["GroupId"], self.port, current_ip_address
                    )
            if not ssh_port_is_open:
                if q.ask(
                    f"Do you want to add a rule to security group
 {sec_group['GroupId']} to allow\n"
```

```
                    f"inbound SSH traffic on port {self.ssh_port} for debugging
 from your computer's IP address of {current_ip_address}? (y/n) ",
                    q.is_yesno,
                ):
                    self.autoscaler.open_inbound_port(
                        sec_group["GroupId"], self.ssh_port, current_ip_address
                    )
            lb_success = self.loadbalancer.verify_load_balancer_endpoint()
        if lb_success:
            print("Your load balancer is ready. You can access it by browsing to:
\n")
            print(f"\thttp://{self.loadbalancer.endpoint()}\n")
        else:
            print(
                "Couldn't get a successful response from the load balancer
 endpoint. Troubleshoot by\n"
                "manually verifying that your VPC and security group are
 configured correctly and that\n"
                "you can successfully make a GET request to the load balancer
 endpoint:\n"
            )
            print(f"\thttp://{self.loadbalancer.endpoint()}\n")
        print("-" * 88)
        q.ask("Press Enter when you're ready to continue with the demo.")

    def demo_choices(self):
        actions = [
            "Send a GET request to the load balancer endpoint.",
            "Check the health of load balancer targets.",
            "Go to the next part of the demo.",
        ]
        choice = 0
        while choice != 2:
            print("-" * 88)
            print(
                "\nSee the current state of the service by selecting one of the
 following choices:\n"
            )
            choice = q.choose("\nWhich action would you like to take? ", actions)
            print("-" * 88)
            if choice == 0:
                print("Request:\n")
                print(f"GET http://{self.loadbalancer.endpoint()}")
                response = requests.get(f"http://{self.loadbalancer.endpoint()}")
```

```
                print("\nResponse:\n")
                print(f"{response.status_code}")
                if response.headers.get("content-type") == "application/json":
                    pp(response.json())
            elif choice == 1:
                print("\nChecking the health of load balancer targets:\n")
                health = self.loadbalancer.check_target_health()
                for target in health:
                    state = target["TargetHealth"]["State"]
                    print(
                        f"\tTarget {target['Target']['Id']} on port
{target['Target']['Port']} is {state}"
                    )
                    if state != "healthy":
                        print(
                            f"\t\t{target['TargetHealth']['Reason']}:
{target['TargetHealth']['Description']}\n"
                        )
                print(
                    f"\nNote that it can take a minute or two for the health
check to update\n"
                    f"after changes are made.\n"
                )
            elif choice == 2:
                print("\nOkay, let's move on.")
                print("-" * 88)

    def demo(self):
        ssm_only_policy = f"{self.resource_path}/ssm_only_policy.json"

        print("\nResetting parameters to starting values for demo.\n")
        self.param_helper.reset()

        print(
            "\nThis part of the demonstration shows how to toggle different parts
of the system\n"
            "to create situations where the web service fails, and shows how
using a resilient\n"
            "architecture can keep the web service running in spite of these
failures."
        )
        print("-" * 88)

        print(
```

```
            "At the start, the load balancer endpoint returns recommendations and
reports that all targets are healthy."
        )
        self.demo_choices()

        print(
            f"The web service running on the EC2 instances gets recommendations
by querying a DynamoDB table.\n"
            f"The table name is contained in a Systems Manager parameter named
'{self.param_helper.table}'.\n"
            f"To simulate a failure of the recommendation service, let's set this
parameter to name a non-existent table.\n"
        )
        self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
        print(
            "\nNow, sending a GET request to the load balancer endpoint returns a
failure code. But, the service reports as\n"
            "healthy to the load balancer because shallow health checks don't
check for failure of the recommendation service."
        )
        self.demo_choices()

        print(
            f"Instead of failing when the recommendation service fails, the web
service can return a static response.\n"
            f"While this is not a perfect solution, it presents the customer with
a somewhat better experience than failure.\n"
        )
        self.param_helper.put(self.param_helper.failure_response, "static")
        print(
            f"\nNow, sending a GET request to the load balancer endpoint returns
a static response.\n"
            f"The service still reports as healthy because health checks are
still shallow.\n"
        )
        self.demo_choices()

        print("Let's reinstate the recommendation service.\n")
        self.param_helper.put(self.param_helper.table,
self.recommendation.table_name)
        print(
            "\nLet's also substitute bad credentials for one of the instances in
the target group so that it can't\n"
            "access the DynamoDB recommendation table.\n"
```

```
        )
        self.autoscaler.create_instance_profile(
            ssm_only_policy,
            self.autoscaler.bad_creds_policy_name,
            self.autoscaler.bad_creds_role_name,
            self.autoscaler.bad_creds_profile_name,
            ["AmazonSSMManagedInstanceCore"],
        )
        instances = self.autoscaler.get_instances()
        bad_instance_id = instances[0]
        instance_profile = self.autoscaler.get_instance_profile(bad_instance_id)
        print(
            f"\nReplacing the profile for instance {bad_instance_id} with a
profile that contains\n"
            f"bad credentials...\n"
        )
        self.autoscaler.replace_instance_profile(
            bad_instance_id,
            self.autoscaler.bad_creds_profile_name,
            instance_profile["AssociationId"],
        )
        print(
            "Now, sending a GET request to the load balancer endpoint returns
either a recommendation or a static response,\n"
            "depending on which instance is selected by the load balancer.\n"
        )
        self.demo_choices()

        print(
            "\nLet's implement a deep health check. For this demo, a deep health
check tests whether\n"
            "the web service can access the DynamoDB table that it depends on for
recommendations. Note that\n"
            "the deep health check is only for ELB routing and not for Auto
Scaling instance health.\n"
            "This kind of deep health check is not recommended for Auto Scaling
instance health, because it\n"
            "risks accidental termination of all instances in the Auto Scaling
group when a dependent service fails.\n"
        )
        print(
            "By implementing deep health checks, the load balancer can detect
when one of the instances is failing\n"
            "and take that instance out of rotation.\n"
```

```
        )
        self.param_helper.put(self.param_helper.health_check, "deep")
        print(
            f"\nNow, checking target health indicates that the instance with bad
credentials ({bad_instance_id})\n"
            f"is unhealthy. Note that it might take a minute or two for the load
balancer to detect the unhealthy \n"
            f"instance. Sending a GET request to the load balancer endpoint
always returns a recommendation, because\n"
            "the load balancer takes unhealthy instances out of its rotation.\n"
        )
        self.demo_choices()

        print(
            "\nBecause the instances in this demo are controlled by an auto
scaler, the simplest way to fix an unhealthy\n"
            "instance is to terminate it and let the auto scaler start a new
instance to replace it.\n"
        )
        self.autoscaler.terminate_instance(bad_instance_id)
        print(
            "\nEven while the instance is terminating and the new instance is
starting, sending a GET\n"
            "request to the web service continues to get a successful
recommendation response because\n"
            "the load balancer routes requests to the healthy instances. After
the replacement instance\n"
            "starts and reports as healthy, it is included in the load balancing
rotation.\n"
            "\nNote that terminating and replacing an instance typically takes
several minutes, during which time you\n"
            "can see the changing health check status until the new instance is
running and healthy.\n"
        )
        self.demo_choices()

        print(
            "\nIf the recommendation service fails now, deep health checks mean
all instances report as unhealthy.\n"
        )
        self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
        print(
            "\nWhen all instances are unhealthy, the load balancer continues to
route requests even to\n"
```

```
            "unhealthy instances, allowing them to fail open and return a static
 response rather than fail\n"
            "closed and report failure to the customer."
        )
        self.demo_choices()
        self.param_helper.reset()

    def destroy(self):
        print(
            "This concludes the demo of how to build and manage a resilient
 service.\n"
            "To keep things tidy and to avoid unwanted charges on your account,
 we can clean up all AWS resources\n"
            "that were created for this demo."
        )
        if q.ask("Do you want to clean up all demo resources? (y/n) ",
 q.is_yesno):
            self.loadbalancer.delete_load_balancer()
            self.loadbalancer.delete_target_group()
            self.autoscaler.delete_group()
            self.autoscaler.delete_key_pair()
            self.autoscaler.delete_template()
            self.autoscaler.delete_instance_profile(
                self.autoscaler.bad_creds_profile_name,
                self.autoscaler.bad_creds_role_name,
            )
            self.recommendation.destroy()
        else:
            print(
                "Okay, we'll leave the resources intact.\n"
                "Don't forget to delete them when you're done with them or you
 might incur unexpected charges."
            )


def main():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--action",
        required=True,
        choices=["all", "deploy", "demo", "destroy"],
        help="The action to take for the demo. When 'all' is specified, resources
 are\n"
        "deployed, the demo is run, and resources are destroyed.",
```

```
    )
    parser.add_argument(
        "--resource_path",
        default="../../../workflows/resilient_service/resources",
        help="The path to resource files used by this example, such as IAM
 policies and\n"
        "instance scripts.",
    )
    args = parser.parse_args()

    print("-" * 88)
    print(
        "Welcome to the demonstration of How to Build and Manage a Resilient
 Service!"
    )
    print("-" * 88)

    prefix = "doc-example-resilience"
    recommendation = RecommendationService.from_client(
        "doc-example-recommendation-service"
    )
    autoscaler = AutoScaler.from_client(prefix)
    loadbalancer = LoadBalancer.from_client(prefix)
    param_helper = ParameterHelper.from_client(recommendation.table_name)
    runner = Runner(
        args.resource_path, recommendation, autoscaler, loadbalancer,
 param_helper
    )
    actions = [args.action] if args.action != "all" else ["deploy", "demo",
"destroy"]
    for action in actions:
        if action == "deploy":
            runner.deploy()
        elif action == "demo":
            runner.demo()
        elif action == "destroy":
            runner.destroy()

    print("-" * 88)
    print("Thanks for watching!")
    print("-" * 88)


if __name__ == "__main__":
```

```
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    main()
```

Crea una classe che racchiuda le operazioni di dimensionamento automatico e Amazon EC2.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
 created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
 that is
                          created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
```

```
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
        self.key_pair_name = f"{resource_prefix}-key-pair"


    @classmethod
    def from_client(cls, resource_prefix):
        """
        Creates this class from Boto3 clients.

        :param resource_prefix: The prefix for naming AWS resources that are
created by this class.
        """
        as_client = boto3.client("autoscaling")
        ec2_client = boto3.client("ec2")
        ssm_client = boto3.client("ssm")
        iam_client = boto3.client("iam")
        return cls(
            resource_prefix,
            "t3.micro",
            "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
            as_client,
            ec2_client,
            ssm_client,
            iam_client,
        )

    def create_instance_profile(
        self, policy_file, policy_name, role_name, profile_name,
aws_managed_policies=()
    ):
        """
        Creates a policy, role, and profile that is associated with instances
created by
        this class. An instance's associated profile defines a role that is
assumed by the
        instance. The role has attached policies that specify the AWS permissions
granted to
        clients that run on the instance.

        :param policy_file: The name of a JSON file that contains the policy
definition to
```

```
                            create and attach to the role.
        :param policy_name: The name to give the created policy.
        :param role_name: The name to give the created role.
        :param profile_name: The name to the created profile.
        :param aws_managed_policies: Additional AWS-managed policies that are
attached to
                                     the role, such as
AmazonSSMManagedInstanceCore to grant
                                     use of Systems Manager to send commands to
the instance.
        :return: The ARN of the profile that is created.
        """
        assume_role_doc = {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {"Service": "ec2.amazonaws.com"},
                    "Action": "sts:AssumeRole",
                }
            ],
        }
        with open(policy_file) as file:
            instance_policy_doc = file.read()

        policy_arn = None
        try:
            pol_response = self.iam_client.create_policy(
                PolicyName=policy_name, PolicyDocument=instance_policy_doc
            )
            policy_arn = pol_response["Policy"]["Arn"]
            log.info("Created policy with ARN %s.", policy_arn)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                log.info("Policy %s already exists, nothing to do.", policy_name)
                list_pol_response = self.iam_client.list_policies(Scope="Local")
                for pol in list_pol_response["Policies"]:
                    if pol["PolicyName"] == policy_name:
                        policy_arn = pol["Arn"]
                        break
            if policy_arn is None:
                raise AutoScalerError(f"Couldn't create policy {policy_name}:
{err}")
```

```
        try:
            self.iam_client.create_role(
                RoleName=role_name,
AssumeRolePolicyDocument=json.dumps(assume_role_doc)
            )
            self.iam_client.attach_role_policy(RoleName=role_name,
PolicyArn=policy_arn)
            for aws_policy in aws_managed_policies:
                self.iam_client.attach_role_policy(
                    RoleName=role_name,
                    PolicyArn=f"arn:aws:iam::aws:policy/{aws_policy}",
                )
            log.info("Created role %s and attached policy %s.", role_name,
policy_arn)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                log.info("Role %s already exists, nothing to do.", role_name)
            else:
                raise AutoScalerError(f"Couldn't create role {role_name}: {err}")

        try:
            profile_response = self.iam_client.create_instance_profile(
                InstanceProfileName=profile_name
            )
            waiter = self.iam_client.get_waiter("instance_profile_exists")
            waiter.wait(InstanceProfileName=profile_name)
            time.sleep(10)  # wait a little longer
            profile_arn = profile_response["InstanceProfile"]["Arn"]
            self.iam_client.add_role_to_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )
            log.info("Created profile %s and added role %s.", profile_name,
role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityAlreadyExists":
                prof_response = self.iam_client.get_instance_profile(
                    InstanceProfileName=profile_name
                )
                profile_arn = prof_response["InstanceProfile"]["Arn"]
                log.info(
                    "Instance profile %s already exists, nothing to do.",
profile_name
                )
            else:
```

```
                raise AutoScalerError(
                    f"Couldn't create profile {profile_name} and attach it to
role\n"
                    f"{role_name}: {err}"
                )
        return profile_arn


    def get_instance_profile(self, instance_id):
        """
        Gets data about the profile associated with an instance.

        :param instance_id: The ID of the instance to look up.
        :return: The profile data.
        """
        try:
            response =
self.ec2_client.describe_iam_instance_profile_associations(
                Filters=[{"Name": "instance-id", "Values": [instance_id]}]
            )
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't get instance profile association for instance
{instance_id}: {err}"
            )
        else:
            return response["IamInstanceProfileAssociations"][0]


    def replace_instance_profile(
        self, instance_id, new_instance_profile_name, profile_association_id
    ):
        """
        Replaces the profile associated with a running instance. After the
profile is
        replaced, the instance is rebooted to ensure that it uses the new
profile. When
        the instance is ready, Systems Manager is used to restart the Python web
server.

        :param instance_id: The ID of the instance to update.
        :param new_instance_profile_name: The name of the new profile to
associate with
                                          the specified instance.
```

```
        :param profile_association_id: The ID of the existing profile association
for the
                                       instance.
        """
        try:
            self.ec2_client.replace_iam_instance_profile_association(
                IamInstanceProfile={"Name": new_instance_profile_name},
                AssociationId=profile_association_id,
            )
            log.info(
                "Replaced instance profile for association %s with profile %s.",
                profile_association_id,
                new_instance_profile_name,
            )
            time.sleep(5)
            inst_ready = False
            tries = 0
            while not inst_ready:
                if tries % 6 == 0:
                    self.ec2_client.reboot_instances(InstanceIds=[instance_id])
                    log.info(
                        "Rebooting instance %s and waiting for it to to be
ready.",
                        instance_id,
                    )
                tries += 1
                time.sleep(10)
                response = self.ssm_client.describe_instance_information()
                for info in response["InstanceInformationList"]:
                    if info["InstanceId"] == instance_id:
                        inst_ready = True
            self.ssm_client.send_command(
                InstanceIds=[instance_id],
                DocumentName="AWS-RunShellScript",
                Parameters={"commands": ["cd / && sudo python3 server.py 80"]},
            )
            log.info("Restarted the Python web server on instance %s.",
instance_id)
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't replace instance profile for association
{profile_association_id}: {err}"
            )
```

```
    def delete_instance_profile(self, profile_name, role_name):
        """
        Detaches a role from an instance profile, detaches policies from the
role,
        and deletes all the resources.

        :param profile_name: The name of the profile to delete.
        :param role_name: The name of the role to delete.
        """
        try:
            self.iam_client.remove_role_from_instance_profile(
                InstanceProfileName=profile_name, RoleName=role_name
            )

self.iam_client.delete_instance_profile(InstanceProfileName=profile_name)
            log.info("Deleted instance profile %s.", profile_name)
            attached_policies = self.iam_client.list_attached_role_policies(
                RoleName=role_name
            )
            for pol in attached_policies["AttachedPolicies"]:
                self.iam_client.detach_role_policy(
                    RoleName=role_name, PolicyArn=pol["PolicyArn"]
                )
                if not pol["PolicyArn"].startswith("arn:aws:iam::aws"):
                    self.iam_client.delete_policy(PolicyArn=pol["PolicyArn"])
                log.info("Detached and deleted policy %s.", pol["PolicyName"])
            self.iam_client.delete_role(RoleName=role_name)
            log.info("Deleted role %s.", role_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "NoSuchEntity":
                log.info(
                    "Instance profile %s doesn't exist, nothing to do.",
profile_name
                )
            else:
                raise AutoScalerError(
                    f"Couldn't delete instance profile {profile_name} or detach "
                    f"policies and delete role {role_name}: {err}"
                )


    def create_key_pair(self, key_pair_name):
        """
```

```
        Creates a new key pair.

        :param key_pair_name: The name of the key pair to create.
        :return: The newly created key pair.
        """
        try:
            response = self.ec2_client.create_key_pair(KeyName=key_pair_name)
            with open(f"{key_pair_name}.pem", "w") as file:
                file.write(response["KeyMaterial"])
            chmod(f"{key_pair_name}.pem", 0o600)
            log.info("Created key pair %s.", key_pair_name)
        except ClientError as err:
            raise AutoScalerError(f"Couldn't create key pair {key_pair_name}:
{err}")


    def delete_key_pair(self):
        """
        Deletes a key pair.

        :param key_pair_name: The name of the key pair to delete.
        """
        try:
            self.ec2_client.delete_key_pair(KeyName=self.key_pair_name)
            remove(f"{self.key_pair_name}.pem")
            log.info("Deleted key pair %s.", self.key_pair_name)
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't delete key pair {self.key_pair_name}: {err}"
            )
        except FileNotFoundError:
            log.info("Key pair %s doesn't exist, nothing to do.",
self.key_pair_name)
        except PermissionError:
            log.info(
                "Inadequate permissions to delete key pair %s.",
self.key_pair_name
            )
        except Exception as err:
            raise AutoScalerError(
                f"Couldn't delete key pair {self.key_pair_name}: {err}"
            )
```

```
    def create_template(self, server_startup_script_file, instance_policy_file):
        """
        Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling. The
        launch template specifies a Bash script in its user data field that runs
after
        the instance is started. This script installs Python packages and starts
a
        Python web server on the instance.

        :param server_startup_script_file: The path to a Bash script file that is
run
                                           when an instance starts.
        :param instance_policy_file: The path to a file that defines a
permissions policy
                                     to create and attach to the instance
profile.
        :return: Information about the newly created template.
        """
        template = {}
        try:
            self.create_key_pair(self.key_pair_name)
            self.create_instance_profile(
                instance_policy_file,
                self.instance_policy_name,
                self.instance_role_name,
                self.instance_profile_name,
            )
            with open(server_startup_script_file) as file:
                start_server_script = file.read()
            ami_latest = self.ssm_client.get_parameter(Name=self.ami_param)
            ami_id = ami_latest["Parameter"]["Value"]
            lt_response = self.ec2_client.create_launch_template(
                LaunchTemplateName=self.launch_template_name,
                LaunchTemplateData={
                    "InstanceType": self.inst_type,
                    "ImageId": ami_id,
                    "IamInstanceProfile": {"Name": self.instance_profile_name},
                    "UserData": base64.b64encode(
                        start_server_script.encode(encoding="utf-8")
                    ).decode(encoding="utf-8"),
                    "KeyName": self.key_pair_name,
                },
            )
```

```
            template = lt_response["LaunchTemplate"]
            log.info(
                "Created launch template %s for AMI %s on %s.",
                self.launch_template_name,
                ami_id,
                self.inst_type,
            )
        except ClientError as err:
            if (
                err.response["Error"]["Code"]
                == "InvalidLaunchTemplateName.AlreadyExistsException"
            ):
                log.info(
                    "Launch template %s already exists, nothing to do.",
                    self.launch_template_name,
                )
            else:
                raise AutoScalerError(
                    f"Couldn't create launch template
{self.launch_template_name}: {err}."
                )
        return template


    def delete_template(self):
        """
        Deletes a launch template.
        """
        try:
            self.ec2_client.delete_launch_template(
                LaunchTemplateName=self.launch_template_name
            )
            self.delete_instance_profile(
                self.instance_profile_name, self.instance_role_name
            )
            log.info("Launch template %s deleted.", self.launch_template_name)
        except ClientError as err:
            if (
                err.response["Error"]["Code"]
                == "InvalidLaunchTemplateName.NotFoundException"
            ):
                log.info(
                    "Launch template %s does not exist, nothing to do.",
                    self.launch_template_name,
```

```
                )
            else:
                raise AutoScalerError(
                    f"Couldn't delete launch template
{self.launch_template_name}: {err}."
                )


    def get_availability_zones(self):
        """
        Gets a list of Availability Zones in the AWS Region of the Amazon EC2
client.

        :return: The list of Availability Zones for the client Region.
        """
        try:
            response = self.ec2_client.describe_availability_zones()
            zones = [zone["ZoneName"] for zone in response["AvailabilityZones"]]
        except ClientError as err:
            raise AutoScalerError(f"Couldn't get availability zones: {err}.")
        else:
            return zones


    def create_group(self, group_size):
        """
        Creates an EC2 Auto Scaling group with the specified size.

        :param group_size: The number of instances to set for the minimum and
maximum in
                           the group.
        :return: The list of Availability Zones specified for the group.
        """
        zones = []
        try:
            zones = self.get_availability_zones()
            self.autoscaling_client.create_auto_scaling_group(
                AutoScalingGroupName=self.group_name,
                AvailabilityZones=zones,
                LaunchTemplate={
                    "LaunchTemplateName": self.launch_template_name,
                    "Version": "$Default",
                },
                MinSize=group_size,
```

```
                MaxSize=group_size,
            )
            log.info(
                "Created EC2 Auto Scaling group %s with availability zones %s.",
                self.launch_template_name,
                zones,
            )
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExists":
                log.info(
                    "EC2 Auto Scaling group %s already exists, nothing to do.",
                    self.group_name,
                )
            else:
                raise AutoScalerError(
                    f"Couldn't create EC2 Auto Scaling group {self.group_name}:
{err}"
                )
        return zones


    def get_instances(self):
        """
        Gets data about the instances in the EC2 Auto Scaling group.

        :return: Data about the instances.
        """
        try:
            as_response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[self.group_name]
            )
            instance_ids = [
                i["InstanceId"]
                for i in as_response["AutoScalingGroups"][0]["Instances"]
            ]
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't get instances for Auto Scaling group
{self.group_name}: {err}"
            )
        else:
            return instance_ids
```

```
    def terminate_instance(self, instance_id):
        """
        Terminates and instances in an EC2 Auto Scaling group. After an instance
is
        terminated, it can no longer be accessed.

        :param instance_id: The ID of the instance to terminate.
        """
        try:
            self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                InstanceId=instance_id, ShouldDecrementDesiredCapacity=False
            )
            log.info("Terminated instance %s.", instance_id)
        except ClientError as err:
            raise AutoScalerError(f"Couldn't terminate instance {instance_id}:
{err}")

    def attach_load_balancer_target_group(self, lb_target_group):
        """
        Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
Scaling group.
        The target group specifies how the load balancer forward requests to the
instances
        in the group.

        :param lb_target_group: Data about the ELB target group to attach.
        """
        try:
            self.autoscaling_client.attach_load_balancer_target_groups(
                AutoScalingGroupName=self.group_name,
                TargetGroupARNs=[lb_target_group["TargetGroupArn"]],
            )
            log.info(
                "Attached load balancer target group %s to auto scaling group
%s.",
                lb_target_group["TargetGroupName"],
                self.group_name,
            )
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't attach load balancer target group
{lb_target_group['TargetGroupName']}\n"
                f"to auto scaling group {self.group_name}"
            )
```

```
   def _try_terminate_instance(self, inst_id):
       stopping = False
       log.info(f"Stopping {inst_id}.")
       while not stopping:
           try:
               self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                   InstanceId=inst_id, ShouldDecrementDesiredCapacity=True
               )
               stopping = True
           except ClientError as err:
               if err.response["Error"]["Code"] == "ScalingActivityInProgress":
                   log.info("Scaling activity in progress for %s. Waiting...",
inst_id)
                   time.sleep(10)
               else:
                   raise AutoScalerError(f"Couldn't stop instance {inst_id}:
{err}.")

   def _try_delete_group(self):
       """
       Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
       the function waits and retries until the group is successfully deleted.
       """
       stopped = False
       while not stopped:
           try:
               self.autoscaling_client.delete_auto_scaling_group(
                   AutoScalingGroupName=self.group_name
               )
               stopped = True
               log.info("Deleted EC2 Auto Scaling group %s.", self.group_name)
           except ClientError as err:
               if (
                   err.response["Error"]["Code"] == "ResourceInUse"
                   or err.response["Error"]["Code"] ==
"ScalingActivityInProgress"
               ):
                   log.info(
                       "Some instances are still running. Waiting for them to
stop..."
                   )
```

```
                    time.sleep(10)
                else:
                    raise AutoScalerError(
                        f"Couldn't delete group {self.group_name}: {err}."
                    )

    def delete_group(self):
        """
        Terminates all instances in the group, deletes the EC2 Auto Scaling
 group.
        """
        try:
            response = self.autoscaling_client.describe_auto_scaling_groups(
                AutoScalingGroupNames=[self.group_name]
            )
            groups = response.get("AutoScalingGroups", [])
            if len(groups) > 0:
                self.autoscaling_client.update_auto_scaling_group(
                    AutoScalingGroupName=self.group_name, MinSize=0
                )
                instance_ids = [inst["InstanceId"] for inst in groups[0]
["Instances"]]
                for inst_id in instance_ids:
                    self._try_terminate_instance(inst_id)
                self._try_delete_group()
            else:
                log.info("No groups found named %s, nothing to do.",
 self.group_name)
        except ClientError as err:
            raise AutoScalerError(f"Couldn't delete group {self.group_name}:
 {err}.")


    def get_default_vpc(self):
        """
        Gets the default VPC for the account.

        :return: Data about the default VPC.
        """
        try:
            response = self.ec2_client.describe_vpcs(
                Filters=[{"Name": "is-default", "Values": ["true"]}]
            )
        except ClientError as err:
```

```
            raise AutoScalerError(f"Couldn't get default VPC: {err}")
        else:
            return response["Vpcs"][0]


    def verify_inbound_port(self, vpc, port, ip_address):
        """
        Verify the default security group of the specified VPC allows ingress
from this
        computer. This can be done by allowing ingress from this computer's IP
        address. In some situations, such as connecting from a corporate network,
you
        must instead specify a prefix list ID. You can also temporarily open the
port to
        any IP address while running this example. If you do, be sure to remove
public
        access when you're done.

        :param vpc: The VPC used by this example.
        :param port: The port to verify.
        :param ip_address: This computer's IP address.
        :return: The default security group of the specific VPC, and a value that
indicates
                 whether the specified port is open.
        """
        try:
            response = self.ec2_client.describe_security_groups(
                Filters=[
                    {"Name": "group-name", "Values": ["default"]},
                    {"Name": "vpc-id", "Values": [vpc["VpcId"]]},
                ]
            )
            sec_group = response["SecurityGroups"][0]
            port_is_open = False
            log.info("Found default security group %s.", sec_group["GroupId"])
            for ip_perm in sec_group["IpPermissions"]:
                if ip_perm.get("FromPort", 0) == port:
                    log.info("Found inbound rule: %s", ip_perm)
                    for ip_range in ip_perm["IpRanges"]:
                        cidr = ip_range.get("CidrIp", "")
                        if cidr.startswith(ip_address) or cidr == "0.0.0.0/0":
                            port_is_open = True
                    if ip_perm["PrefixListIds"]:
                        port_is_open = True
```

```
                    if not port_is_open:
                        log.info(
                            "The inbound rule does not appear to be open to
either this computer's IP\n"
                            "address of %s, to all IP addresses (0.0.0.0/0), or
to a prefix list ID.",
                            ip_address,
                        )
                    else:
                        break
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't verify inbound rule for port {port} for VPC
{vpc['VpcId']}: {err}"
            )
        else:
            return sec_group, port_is_open


    def open_inbound_port(self, sec_group_id, port, ip_address):
        """
        Add an ingress rule to the specified security group that allows access on
the
        specified port from the specified IP address.

        :param sec_group_id: The ID of the security group to modify.
        :param port: The port to open.
        :param ip_address: The IP address that is granted access.
        """
        try:
            self.ec2_client.authorize_security_group_ingress(
                GroupId=sec_group_id,
                CidrIp=f"{ip_address}/32",
                FromPort=port,
                ToPort=port,
                IpProtocol="tcp",
            )
            log.info(
                "Authorized ingress to %s on port %s from %s.",
                sec_group_id,
                port,
                ip_address,
            )
        except ClientError as err:
```

```
            raise AutoScalerError(
                f"Couldn't authorize ingress to {sec_group_id} on port {port}
 from {ip_address}: {err}"
            )


    def get_subnets(self, vpc_id, zones):
        """
        Gets the default subnets in a VPC for a specified list of Availability
 Zones.

        :param vpc_id: The ID of the VPC to look up.
        :param zones: The list of Availability Zones to look up.
        :return: The list of subnets found.
        """
        try:
            response = self.ec2_client.describe_subnets(
                Filters=[
                    {"Name": "vpc-id", "Values": [vpc_id]},
                    {"Name": "availability-zone", "Values": zones},
                    {"Name": "default-for-az", "Values": ["true"]},
                ]
            )
            subnets = response["Subnets"]
            log.info("Found %s subnets for the specified zones.", len(subnets))
        except ClientError as err:
            raise AutoScalerError(f"Couldn't get subnets: {err}")
        else:
            return subnets
```

Crea una classe che racchiuda le operazioni di Elastic Load Balancing.

```
class LoadBalancer:
    """Encapsulates Elastic Load Balancing (ELB) actions."""

    def __init__(self, target_group_name, load_balancer_name, elb_client):
        """
        :param target_group_name: The name of the target group associated with
 the load balancer.
```

```
        :param load_balancer_name: The name of the load balancer.
        :param elb_client: A Boto3 Elastic Load Balancing client.
        """
        self.target_group_name = target_group_name
        self.load_balancer_name = load_balancer_name
        self.elb_client = elb_client
        self._endpoint = None


    @classmethod
    def from_client(cls, resource_prefix):
        """
        Creates this class from a Boto3 client.

        :param resource_prefix: The prefix to give to AWS resources created by
this class.
        """
        elb_client = boto3.client("elbv2")
        return cls(f"{resource_prefix}-tg", f"{resource_prefix}-lb", elb_client)

    def endpoint(self):
        """
        Gets the HTTP endpoint of the load balancer.

        :return: The endpoint.
        """
        if self._endpoint is None:
            try:
                response = self.elb_client.describe_load_balancers(
                    Names=[self.load_balancer_name]
                )
                self._endpoint = response["LoadBalancers"][0]["DNSName"]
            except ClientError as err:
                raise LoadBalancerError(
                    f"Couldn't get the endpoint for load balancer
{self.load_balancer_name}: {err}"
                )
        return self._endpoint


    def create_target_group(self, protocol, port, vpc_id):
        """
        Creates an Elastic Load Balancing target group. The target group
specifies how
```

```
        the load balancer forward requests to instances in the group and how
instance
        health is checked.

        To speed up this demo, the health check is configured with shortened
times and
        lower thresholds. In production, you might want to decrease the
sensitivity of
        your health checks to avoid unwanted failures.

        :param protocol: The protocol to use to forward requests, such as 'HTTP'.
        :param port: The port to use to forward requests, such as 80.
        :param vpc_id: The ID of the VPC in which the load balancer exists.
        :return: Data about the newly created target group.
        """
        try:
            response = self.elb_client.create_target_group(
                Name=self.target_group_name,
                Protocol=protocol,
                Port=port,
                HealthCheckPath="/healthcheck",
                HealthCheckIntervalSeconds=10,
                HealthCheckTimeoutSeconds=5,
                HealthyThresholdCount=2,
                UnhealthyThresholdCount=2,
                VpcId=vpc_id,
            )
            target_group = response["TargetGroups"][0]
            log.info("Created load balancing target group %s.",
self.target_group_name)
        except ClientError as err:
            raise LoadBalancerError(
                f"Couldn't create load balancing target group
{self.target_group_name}: {err}"
            )
        else:
            return target_group


    def delete_target_group(self):
        """
        Deletes the target group.
        """
        done = False
```

```
        while not done:
            try:
                response = self.elb_client.describe_target_groups(
                    Names=[self.target_group_name]
                )
                tg_arn = response["TargetGroups"][0]["TargetGroupArn"]
                self.elb_client.delete_target_group(TargetGroupArn=tg_arn)
                log.info(
                    "Deleted load balancing target group %s.",
self.target_group_name
                )
                done = True
            except ClientError as err:
                if err.response["Error"]["Code"] == "TargetGroupNotFound":
                    log.info(
                        "Load balancer target group %s not found, nothing to
do.",
                        self.target_group_name,
                    )
                    done = True
                elif err.response["Error"]["Code"] == "ResourceInUse":
                    log.info(
                        "Target group not yet released from load balancer,
waiting..."
                    )
                    time.sleep(10)
                else:
                    raise LoadBalancerError(
                        f"Couldn't delete load balancing target group
{self.target_group_name}: {err}"
                    )


    def create_load_balancer(self, subnet_ids, target_group):
        """
        Creates an Elastic Load Balancing load balancer that uses the specified
subnets
        and forwards requests to the specified target group.

        :param subnet_ids: A list of subnets to associate with the load balancer.
        :param target_group: An existing target group that is added as a listener
to the
                             load balancer.
        :return: Data about the newly created load balancer.
```

```
        """
        try:
            response = self.elb_client.create_load_balancer(
                Name=self.load_balancer_name, Subnets=subnet_ids
            )
            load_balancer = response["LoadBalancers"][0]
            log.info("Created load balancer %s.", self.load_balancer_name)
            waiter = self.elb_client.get_waiter("load_balancer_available")
            log.info("Waiting for load balancer to be available...")
            waiter.wait(Names=[self.load_balancer_name])
            log.info("Load balancer is available!")
            self.elb_client.create_listener(
                LoadBalancerArn=load_balancer["LoadBalancerArn"],
                Protocol=target_group["Protocol"],
                Port=target_group["Port"],
                DefaultActions=[
                    {
                        "Type": "forward",
                        "TargetGroupArn": target_group["TargetGroupArn"],
                    }
                ],
            )
            log.info(
                "Created listener to forward traffic from load balancer %s to
target group %s.",
                self.load_balancer_name,
                target_group["TargetGroupName"],
            )
        except ClientError as err:
            raise LoadBalancerError(
                f"Failed to create load balancer {self.load_balancer_name}"
                f"and add a listener for target group
{target_group['TargetGroupName']}: {err}"
            )
        else:
            self._endpoint = load_balancer["DNSName"]
            return load_balancer


    def delete_load_balancer(self):
        """
        Deletes a load balancer.
        """
        try:
```

```
            response = self.elb_client.describe_load_balancers(
                Names=[self.load_balancer_name]
            )
            lb_arn = response["LoadBalancers"][0]["LoadBalancerArn"]
            self.elb_client.delete_load_balancer(LoadBalancerArn=lb_arn)
            log.info("Deleted load balancer %s.", self.load_balancer_name)
            waiter = self.elb_client.get_waiter("load_balancers_deleted")
            log.info("Waiting for load balancer to be deleted...")
            waiter.wait(Names=[self.load_balancer_name])
        except ClientError as err:
            if err.response["Error"]["Code"] == "LoadBalancerNotFound":
                log.info(
                    "Load balancer %s does not exist, nothing to do.",
                    self.load_balancer_name,
                )
            else:
                raise LoadBalancerError(
                    f"Couldn't delete load balancer {self.load_balancer_name}:
{err}"
                )


    def verify_load_balancer_endpoint(self):
        """
        Verify this computer can successfully send a GET request to the load
balancer endpoint.
        """
        success = False
        retries = 3
        while not success and retries > 0:
            try:
                lb_response = requests.get(f"http://{self.endpoint()}")
                log.info(
                    "Got response %s from load balancer endpoint.",
                    lb_response.status_code,
                )
                if lb_response.status_code == 200:
                    success = True
                else:
                    retries = 0
            except requests.exceptions.ConnectionError:
                log.info(
                    "Got connection error from load balancer endpoint,
retrying..."
```

```
                )
                retries -= 1
                time.sleep(10)
        return success

    def check_target_health(self):
        """
        Checks the health of the instances in the target group.

        :return: The health status of the target group.
        """
        try:
            tg_response = self.elb_client.describe_target_groups(
                Names=[self.target_group_name]
            )
            health_response = self.elb_client.describe_target_health(
                TargetGroupArn=tg_response["TargetGroups"][0]["TargetGroupArn"]
            )
        except ClientError as err:
            raise LoadBalancerError(
                f"Couldn't check health of {self.target_group_name} targets:
 {err}"
            )
        else:
            return health_response["TargetHealthDescriptions"]
```

Crea una classe che utilizzi DynamoDB per simulare un servizio di raccomandazione.

```
class RecommendationService:
    """
    Encapsulates a DynamoDB table to use as a service that recommends books,
 movies,
    and songs.
    """

    def __init__(self, table_name, dynamodb_client):
        """
        :param table_name: The name of the DynamoDB recommendations table.
        :param dynamodb_client: A Boto3 DynamoDB client.
```

```
        """
        self.table_name = table_name
        self.dynamodb_client = dynamodb_client

    @classmethod
    def from_client(cls, table_name):
        """
        Creates this class from a Boto3 client.

        :param table_name: The name of the DynamoDB recommendations table.
        """
        ddb_client = boto3.client("dynamodb")
        return cls(table_name, ddb_client)

    def create(self):
        """
        Creates a DynamoDB table to use a recommendation service. The table has a
        hash key named 'MediaType' that defines the type of media recommended,
such as
        Book or Movie, and a range key named 'ItemId' that, combined with the
MediaType,
        forms a unique identifier for the recommended item.

        :return: Data about the newly created table.
        """
        try:
            response = self.dynamodb_client.create_table(
                TableName=self.table_name,
                AttributeDefinitions=[
                    {"AttributeName": "MediaType", "AttributeType": "S"},
                    {"AttributeName": "ItemId", "AttributeType": "N"},
                ],
                KeySchema=[
                    {"AttributeName": "MediaType", "KeyType": "HASH"},
                    {"AttributeName": "ItemId", "KeyType": "RANGE"},
                ],
                ProvisionedThroughput={"ReadCapacityUnits": 5,
"WriteCapacityUnits": 5},
            )
            log.info("Creating table %s...", self.table_name)
            waiter = self.dynamodb_client.get_waiter("table_exists")
            waiter.wait(TableName=self.table_name)
            log.info("Table %s created.", self.table_name)
        except ClientError as err:
```

```
            if err.response["Error"]["Code"] == "ResourceInUseException":
                log.info("Table %s exists, nothing to be do.", self.table_name)
            else:
                raise RecommendationServiceError(
                    self.table_name, f"ClientError when creating table: {err}."
                )
        else:
            return response

    def populate(self, data_file):
        """
        Populates the recommendations table from a JSON file.

        :param data_file: The path to the data file.
        """
        try:
            with open(data_file) as data:
                items = json.load(data)
            batch = [{"PutRequest": {"Item": item}} for item in items]
            self.dynamodb_client.batch_write_item(RequestItems={self.table_name:
batch})
            log.info(
                "Populated table %s with items from %s.", self.table_name,
data_file
            )
        except ClientError as err:
            raise RecommendationServiceError(
                self.table_name, f"Couldn't populate table from {data_file}:
{err}"
            )

    def destroy(self):
        """
        Deletes the recommendations table.
        """
        try:
            self.dynamodb_client.delete_table(TableName=self.table_name)
            log.info("Deleting table %s...", self.table_name)
            waiter = self.dynamodb_client.get_waiter("table_not_exists")
            waiter.wait(TableName=self.table_name)
            log.info("Table %s deleted.", self.table_name)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
```

```
                log.info("Table %s does not exist, nothing to do.",
 self.table_name)
            else:
                raise RecommendationServiceError(
                    self.table_name, f"ClientError when deleting table: {err}."
                )
```

Crea una classe che racchiuda le operazioni di Systems Manager.

```
class ParameterHelper:
    """
    Encapsulates Systems Manager parameters. This example uses these parameters
 to drive
    the demonstration of resilient architecture, such as failure of a dependency
 or
    how the service responds to a health check.
    """

    table = "doc-example-resilient-architecture-table"
    failure_response = "doc-example-resilient-architecture-failure-response"
    health_check = "doc-example-resilient-architecture-health-check"

    def __init__(self, table_name, ssm_client):
        """
        :param table_name: The name of the DynamoDB table that is used as a
 recommendation
                           service.
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.table_name = table_name

    @classmethod
    def from_client(cls, table_name):
        ssm_client = boto3.client("ssm")
        return cls(table_name, ssm_client)

    def reset(self):
        """
        Resets the Systems Manager parameters to starting values for the demo.
```

```
        These are the name of the DynamoDB recommendation table, no response when
a
        dependency fails, and shallow health checks.
        """
        self.put(self.table, self.table_name)
        self.put(self.failure_response, "none")
        self.put(self.health_check, "shallow")

    def put(self, name, value):
        """
        Sets the value of a named Systems Manager parameter.

        :param name: The name of the parameter.
        :param value: The new value of the parameter.
        """
        try:
            self.ssm_client.put_parameter(
                Name=name, Value=value, Overwrite=True, Type="String"
            )
            log.info("Setting demo parameter %s to '%s'.", name, value)
        except ClientError as err:
            raise ParameterHelperError(
                f"Couldn't set parameter {name} to {value}: {err}"
            )
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AttachLoadBalancerTargetGruppi
  - CreateAutoScalingGroup
  - CreateInstanceProfilo
  - CreateLaunchModello
  - CreateListener
  - CreateLoadBilanciatore
  - CreateTargetGruppo
  - DeleteAutoScalingGroup
  - DeleteInstanceProfilo

- DeleteLaunchModello
- DeleteLoadBilanciatore
- DeleteTargetGruppo
- DescribeAutoScalingGroups
- DescribeAvailabilityZone
- DescribeIamInstanceProfileAssociazioni
- DescribeInstances
- DescribeLoadBilanciatori
- DescribeSubnets
- DescribeTargetGruppi
- DescribeTargetHealth
- DescribeVpcs
- RebootInstances
- ReplaceIamInstanceProfileAssociazione
- TerminateInstanceInAutoScalingGroup
- UpdateAutoScalingGroup

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un gruppo IAM e aggiungi un utente al gruppo utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Crea un gruppo e concedigli autorizzazioni di accesso complete ad Amazon S3.
- Crea un nuovo utente senza autorizzazioni per accedere ad Amazon S3.
- Aggiungi l'utente al gruppo e verifica che ora disponga delle autorizzazioni per Amazon S3, quindi ripulisci le risorse.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
global using Amazon.IdentityManagement;
global using Amazon.S3;
global using Amazon.SecurityToken;
global using IAMActions;
global using IamScenariosCommon;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;


namespace IAMActions;

public class IAMWrapper
{
    private readonly IAmazonIdentityManagementService _IAMService;

    /// <summary>
    /// Constructor for the IAMWrapper class.
    /// </summary>
    /// <param name="IAMService">An IAM client object.</param>
    public IAMWrapper(IAmazonIdentityManagementService IAMService)
    {
        _IAMService = IAMService;
    }

    /// <summary>
    /// Add an existing IAM user to an existing IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to add.</param>
    /// <param name="groupName">The name of the group to add the user to.</param>
```

```
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AddUserToGroupAsync(string userName, string
 groupName)
    {
        var response = await _IAMService.AddUserToGroupAsync(new
 AddUserToGroupRequest
        {
            GroupName = groupName,
            UserName = userName,
        });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
```

```
        var response = await _IAMService.CreateAccessKeyAsync(new
 CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }


    /// <summary>
    /// Create an IAM group.
    /// </summary>
    /// <param name="groupName">The name to give the IAM group.</param>
    /// <returns>The IAM group that was created.</returns>
    public async Task<Group> CreateGroupAsync(string groupName)
    {
        var response = await _IAMService.CreateGroupAsync(new CreateGroupRequest
 { GroupName = groupName });
        return response.Group;
    }


    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });

        return response.Policy;
    }
```

```
    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
 rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }


    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
    }


    /// <summary>
```

```
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupAsync(string groupName)
    {
        var response = await _IAMService.DeleteGroupAsync(new DeleteGroupRequest
{ GroupName = groupName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

```
   /// <summary>
   /// Delete an IAM policy associated with an IAM group.
   /// </summary>
   /// <param name="groupName">The name of the IAM group associated with the
   /// policy.</param>
   /// <param name="policyName">The name of the policy to delete.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DeleteGroupPolicyAsync(string groupName, string
policyName)
   {
       var request = new DeleteGroupPolicyRequest()
       {
           GroupName = groupName,
           PolicyName = policyName,
       };

       var response = await _IAMService.DeleteGroupPolicyAsync(request);
       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }


   /// <summary>
   /// Delete an IAM policy.
   /// </summary>
   /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
   /// delete.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DeletePolicyAsync(string policyArn)
   {
       var response = await _IAMService.DeletePolicyAsync(new
DeletePolicyRequest { PolicyArn = policyArn });
       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }


   /// <summary>
   /// Delete an IAM role.
   /// </summary>
   /// <param name="roleName">The name of the IAM role to delete.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DeleteRoleAsync(string roleName)
   {
```

```
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
 { UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user policy.
    /// </summary>
```

```
    /// <param name="policyName">The name of the IAM policy to delete.</param>
    /// <param name="userName">The username of the IAM user.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
    {
        var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Detach an IAM policy from an IAM role.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
    {
        var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Gets the IAM password policy for an AWS account.
    /// </summary>
    /// <returns>The PasswordPolicy for the AWS account.</returns>
    public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
    {
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }
```

```
    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }


    /// <summary>
    /// Get information about an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to retrieve information
    /// for.</param>
    /// <returns>The IAM role that was retrieved.</returns>
    public async Task<Role> GetRoleAsync(string roleName)
    {
        var response = await _IAMService.GetRoleAsync(new GetRoleRequest
        {
            RoleName = roleName,
        });

        return response.Role;
    }


    /// <summary>
    /// Get information about an IAM user.
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
 { UserName = userName });
        return response.User;
```

```
    }


    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }


    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }

        return groups;
    }
```

```
    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }


    /// <summary>
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
 _IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
 roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }


    /// <summary>
    /// List IAM roles.
    /// </summary>
```

```
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }


    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }


    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
```

```
    }


    /// <summary>
    /// Remove a user from an IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to remove.</param>
    /// <param name="groupName">The name of the IAM group to remove the user
from.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> RemoveUserFromGroupAsync(string userName, string
groupName)
    {
        // Remove the user from the group.
        var removeUserRequest = new RemoveUserFromGroupRequest()
        {
            UserName = userName,
            GroupName = groupName,
        };

        var response = await
_IAMService.RemoveUserFromGroupAsync(removeUserRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutGroupPolicyAsync(string groupName, string
policyName, string policyDocument)
    {
        var request = new PutGroupPolicyRequest
        {
            GroupName = groupName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };
```

```
        var response = await _IAMService.PutGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM user.
    /// </summary>
    /// <param name="userName">The name of the IAM user.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
 policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutUserPolicyAsync(string userName, string
 policyName, string policyDocument)
    {
        var request = new PutUserPolicyRequest
        {
            UserName = userName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
```

```
        };

        var response = await _IAMService.PutUserPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

    /// <summary>
    /// Wait for a new access key to be ready to use.
    /// </summary>
    /// <param name="accessKeyId">The Id of the access key.</param>
    /// <returns>A boolean value indicating the success of the action.</returns>
    public async Task<bool> WaitUntilAccessKeyIsReady(string accessKeyId)
    {
        var keyReady = false;

        do
        {
            try
            {
                var response = await _IAMService.GetAccessKeyLastUsedAsync(
                    new GetAccessKeyLastUsedRequest { AccessKeyId =
 accessKeyId });
                if (response.UserName is not null)
                {
                    keyReady = true;
                }
            }
            catch (NoSuchEntityException)
            {
                keyReady = false;
            }
        } while (!keyReady);

        return keyReady;
    }
}


using Microsoft.Extensions.Configuration;

namespace IAMGroups;

public class IAMGroups
{
```

```
    private static ILogger logger = null!;

    // Represents JSON code for AWS full access policy for Amazon Simple
    // Storage Service (Amazon S3).
    private const string S3FullAccessPolicyDocument = "{" +
        " \"Statement\" : [{" +
            " \"Action\" : [\"s3:*\"]," +
            " \"Effect\" : \"Allow\"," +
            " \"Resource\" : \"*\"" +
        "}]" +
    "}";


    static async Task Main(string[] args)
    {
        // Set up dependency injection for the AWS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
            .AddTransient<IAMWrapper>()
            .AddTransient<UIWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<IAMGroups>();

        IConfiguration configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        var groupUserName = configuration["GroupUserName"];
        var groupName = configuration["GroupName"];
        var groupPolicyName = configuration["GroupPolicyName"];
        var groupBucketName = configuration["GroupBucketName"];
```

```
        var wrapper = host.Services.GetRequiredService<IAMWrapper>();
        var uiWrapper = host.Services.GetRequiredService<UIWrapper>();

        uiWrapper.DisplayGroupsOverview();
        uiWrapper.PressEnter();

        // Create an IAM group.
        uiWrapper.DisplayTitle("Create IAM group");
        Console.WriteLine("Let's begin by creating a new IAM group.");
        var group = await wrapper.CreateGroupAsync(groupName);

        // Add an inline IAM policy to the group.
        uiWrapper.DisplayTitle("Add policy to group");
        Console.WriteLine("Add an inline policy to the group that allows members
to have full access to");
        Console.WriteLine("Amazon Simple Storage Service (Amazon S3) buckets.");

        await wrapper.PutGroupPolicyAsync(group.GroupName, groupPolicyName,
S3FullAccessPolicyDocument);

        uiWrapper.PressEnter();

        // Now create a new user.
        uiWrapper.DisplayTitle("Create an IAM user");
        Console.WriteLine("Now let's create a new IAM user.");
        var groupUser = await wrapper.CreateUserAsync(groupUserName);

        // Add the new user to the group.
        uiWrapper.DisplayTitle("Add the user to the group");
        Console.WriteLine("Adding the user to the group, which will give the user
the same permissions as the group.");
        await wrapper.AddUserToGroupAsync(groupUser.UserName, group.GroupName);

        Console.WriteLine($"User, {groupUser.UserName}, has been added to the
group, {group.GroupName}.");
        uiWrapper.PressEnter();

        Console.WriteLine("Now that we have created a user, and added the user to
the group, let's create an IAM access key.");

        // Create access and secret keys for the user.
        var accessKey = await wrapper.CreateAccessKeyAsync(groupUserName);
        Console.WriteLine("Key created.");
```

```
        uiWrapper.WaitABit(15, "Waiting for the access key to be ready for
use.");

        uiWrapper.DisplayTitle("List buckets");
        Console.WriteLine("To prove that the user has access to Amazon S3, list
the S3 buckets for the account.");

        var s3Client = new AmazonS3Client(accessKey.AccessKeyId,
accessKey.SecretAccessKey);
        var stsClient = new
AmazonSecurityTokenServiceClient(accessKey.AccessKeyId,
accessKey.SecretAccessKey);

        var s3Wrapper = new S3Wrapper(s3Client, stsClient);

        var buckets = await s3Wrapper.ListMyBucketsAsync();

        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
            {
                Console.WriteLine($"{bucket.BucketName}\tcreated on:
{bucket.CreationDate}");
            });
        }

        // Show that the user also has write access to Amazon S3 by creating
        // a new bucket.
        uiWrapper.DisplayTitle("Create a bucket");
        Console.WriteLine("Since group members have full access to Amazon S3,
let's create a bucket.");
        var success = await s3Wrapper.PutBucketAsync(groupBucketName);

        if (success)
        {
            Console.WriteLine($"Successfully created the bucket:
{groupBucketName}.");
        }

        uiWrapper.PressEnter();

        Console.WriteLine("Let's list the user's S3 buckets again to show the new
bucket.");
```

```
        buckets = await s3Wrapper.ListMyBucketsAsync();

        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
            {
                Console.WriteLine($"{bucket.BucketName}\tcreated on:
 {bucket.CreationDate}");
            });
        }

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Clean up resources");
        Console.WriteLine("First delete the bucket we created.");
        await s3Wrapper.DeleteBucketAsync(groupBucketName);

        Console.WriteLine($"Now remove the user, {groupUserName}, from the group,
 {groupName}.");
        await wrapper.RemoveUserFromGroupAsync(groupUserName, groupName);

        Console.WriteLine("Delete the user's access key.");
        await wrapper.DeleteAccessKeyAsync(accessKey.AccessKeyId, groupUserName);

        // Now we can safely delete the user.
        Console.WriteLine("Now we can delete the user.");
        await wrapper.DeleteUserAsync(groupUserName);

        uiWrapper.PressEnter();

        Console.WriteLine("Now we will delete the IAM policy attached to the
 group.");
        await wrapper.DeleteGroupPolicyAsync(groupName, groupPolicyName);

        Console.WriteLine("Now we delete the IAM group.");
        await wrapper.DeleteGroupAsync(groupName);

        uiWrapper.PressEnter();

        Console.WriteLine("The IAM groups demo has completed.");

        uiWrapper.PressEnter();
    }
}
```

```
namespace IamScenariosCommon;

using System.Net;

/// <summary>
/// A class to perform Amazon Simple Storage Service (Amazon S3) actions for
/// the IAM Basics scenario.
/// </summary>
public class S3Wrapper
{
    private IAmazonS3 _s3Service;
    private IAmazonSecurityTokenService _stsService;

    /// <summary>
    /// Constructor for the S3Wrapper class.
    /// </summary>
    /// <param name="s3Service">An Amazon S3 client object.</param>
    /// <param name="stsService">An AWS Security Token Service (AWS STS)
    /// client object.</param>
    public S3Wrapper(IAmazonS3 s3Service, IAmazonSecurityTokenService stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }

    /// <summary>
    /// Assumes an AWS Identity and Access Management (IAM) role that allows
    /// Amazon S3 access for the current session.
    /// </summary>
    /// <param name="roleSession">A string representing the current session.</
param>
    /// <param name="roleToAssume">The name of the IAM role to assume.</param>
    /// <returns>Credentials for the newly assumed IAM role.</returns>
    public async Task<Credentials> AssumeS3RoleAsync(string roleSession, string
 roleToAssume)
    {
        // Create the request to use with the AssumeRoleAsync call.
        var request = new AssumeRoleRequest()
        {
            RoleSessionName = roleSession,
            RoleArn = roleToAssume,
        };
```

```
        var response = await _stsService.AssumeRoleAsync(request);

        return response.Credentials;
    }


    /// <summary>
    /// Delete an S3 bucket.
    /// </summary>
    /// <param name="bucketName">Name of the S3 bucket to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteBucketAsync(string bucketName)
    {
        var result = await _s3Service.DeleteBucketAsync(new DeleteBucketRequest
{ BucketName = bucketName });
        return result.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the buckets that are owned by the user's account.
    /// </summary>
    /// <returns>Async Task.</returns>
    public async Task<List<S3Bucket>?> ListMyBucketsAsync()
    {
        try
        {
            // Get the list of buckets accessible by the new user.
            var response = await _s3Service.ListBucketsAsync();

            return response.Buckets;
        }
        catch (AmazonS3Exception ex)
        {
            // Something else went wrong. Display the error message.
            Console.WriteLine($"Error: {ex.Message}");
            return null;
        }
    }

    /// <summary>
    /// Create a new S3 bucket.
    /// </summary>
    /// <param name="bucketName">The name for the new bucket.</param>
```

```
    /// <returns>A Boolean value indicating whether the action completed
    /// successfully.</returns>
    public async Task<bool> PutBucketAsync(string bucketName)
    {
        var response = await _s3Service.PutBucketAsync(new PutBucketRequest
 { BucketName = bucketName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Update the client objects with new client objects. This is available
    /// because the scenario uses the methods of this class without and then
    /// with the proper permissions to list S3 buckets.
    /// </summary>
    /// <param name="s3Service">The Amazon S3 client object.</param>
    /// <param name="stsService">The AWS STS client object.</param>
    public void UpdateClients(IAmazonS3 s3Service, IAmazonSecurityTokenService
 stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }
}


namespace IamScenariosCommon;

public class UIWrapper
{
    public readonly string SepBar = new('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the IAM Groups scenario.
    /// </summary>
    public void DisplayGroupsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to the IAM Groups Demo");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates an Amazon Identity and Access Management
 (IAM) group.");
        Console.WriteLine("\t2. Adds an IAM policy to the IAM group giving it
 full access to Amazon S3.");
```

```
        Console.WriteLine("\t3. Creates a new IAM user.");
        Console.WriteLine("\t4. Creates an IAM access key for the user.");
        Console.WriteLine("\t5. Adds the user to the IAM group.");
        Console.WriteLine("\t6. Lists the buckets on the account.");
        Console.WriteLine("\t7. Proves that the user has full Amazon S3 access by
creating a bucket.");
        Console.WriteLine("\t8. List the buckets again to show the new bucket.");
        Console.WriteLine("\t9. Cleans up all the resources created.");
    }

    /// <summary>
    /// Show information about the IAM Basics scenario.
    /// </summary>
    public void DisplayBasicsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to IAM Basics");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates a user with no permissions.");
        Console.WriteLine("\t2. Creates a role and policy that grant
s3:ListAllMyBuckets permission.");
        Console.WriteLine("\t3. Grants the user permission to assume the role.");
        Console.WriteLine("\t4. Creates an S3 client object as the user and tries
to list buckets (this will fail).");
        Console.WriteLine("\t5. Gets temporary credentials by assuming the
role.");
        Console.WriteLine("\t6. Creates a new S3 client object with the temporary
credentials and lists the buckets (this will succeed).");
        Console.WriteLine("\t7. Deletes all the resources.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPress <Enter> to continue. ");
        _ = Console.ReadLine();
        Console.WriteLine();
    }

    /// <summary>
    /// Pad a string with spaces to center it on the console display.
```

```
    /// </summary>
    /// <param name="strToCenter">The string to be centered.</param>
    /// <returns>The padded string.</returns>
    public string CenterString(string strToCenter)
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title, and another
    /// line of hyphens.
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }

    /// <summary>
    /// Display a countdown and wait for a number of seconds.
    /// </summary>
    /// <param name="numSeconds">The number of seconds to wait.</param>
    public void WaitABit(int numSeconds, string msg)
    {
        Console.WriteLine(msg);

        // Wait for the requested number of seconds.
        for (int i = numSeconds; i > 0; i--)
        {
            System.Threading.Thread.Sleep(1000);
            Console.Write($"{i}...");
        }

        PressEnter();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - AddUserToGroup
  - AttachRolePolitica
  - CreateAccessChiave
  - CreateGroup
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeleteGroup
  - DeleteGroupPolitica
  - DeleteUser
  - PutGroupPolitica
  - RemoveUserFromGroup

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un utente IAM e assumi un ruolo AWS STS utilizzando un AWS SDK

Gli esempi di codice seguenti mostrano come creare un utente e assumere un ruolo.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Crea un utente che non disponga di autorizzazioni.
- Crea un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3 per l'account.
- Aggiungi una policy per consentire all'utente di assumere il ruolo.

- Assumi il ruolo ed elenca i bucket S3 utilizzando le credenziali temporanee, quindi ripulisci le risorse.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
global using Amazon.IdentityManagement;
global using Amazon.S3;
global using Amazon.SecurityToken;
global using IAMActions;
global using IamScenariosCommon;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;


namespace IAMActions;

public class IAMWrapper
{
    private readonly IAmazonIdentityManagementService _IAMService;

    /// <summary>
    /// Constructor for the IAMWrapper class.
    /// </summary>
    /// <param name="IAMService">An IAM client object.</param>
    public IAMWrapper(IAmazonIdentityManagementService IAMService)
    {
        _IAMService = IAMService;
    }

    /// <summary>
```

```
    /// Add an existing IAM user to an existing IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to add.</param>
    /// <param name="groupName">The name of the group to add the user to.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AddUserToGroupAsync(string userName, string
 groupName)
    {
        var response = await _IAMService.AddUserToGroupAsync(new
 AddUserToGroupRequest
        {
            GroupName = groupName,
            UserName = userName,
        });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Attach an IAM policy to a role.
    /// </summary>
    /// <param name="policyArn">The policy to attach.</param>
    /// <param name="roleName">The role that the policy will be attached to.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> AttachRolePolicyAsync(string policyArn, string
 roleName)
    {
        var response = await _IAMService.AttachRolePolicyAsync(new
 AttachRolePolicyRequest
        {
            PolicyArn = policyArn,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an IAM access key for a user.
    /// </summary>
    /// <param name="userName">The username for which to create the IAM access
```

```
    /// key.</param>
    /// <returns>The AccessKey.</returns>
    public async Task<AccessKey> CreateAccessKeyAsync(string userName)
    {
        var response = await _IAMService.CreateAccessKeyAsync(new
 CreateAccessKeyRequest
        {
            UserName = userName,
        });

        return response.AccessKey;

    }


    /// <summary>
    /// Create an IAM group.
    /// </summary>
    /// <param name="groupName">The name to give the IAM group.</param>
    /// <returns>The IAM group that was created.</returns>
    public async Task<Group> CreateGroupAsync(string groupName)
    {
        var response = await _IAMService.CreateGroupAsync(new CreateGroupRequest
 { GroupName = groupName });
        return response.Group;
    }


    /// <summary>
    /// Create an IAM policy.
    /// </summary>
    /// <param name="policyName">The name to give the new IAM policy.</param>
    /// <param name="policyDocument">The policy document for the new policy.</
param>
    /// <returns>The new IAM policy object.</returns>
    public async Task<ManagedPolicy> CreatePolicyAsync(string policyName, string
 policyDocument)
    {
        var response = await _IAMService.CreatePolicyAsync(new
 CreatePolicyRequest
        {
            PolicyDocument = policyDocument,
            PolicyName = policyName,
        });
```

```
        return response.Policy;
    }


    /// <summary>
    /// Create a new IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="rolePolicyDocument">The name of the IAM policy document
    /// for the new role.</param>
    /// <returns>The Amazon Resource Name (ARN) of the role.</returns>
    public async Task<string> CreateRoleAsync(string roleName, string
 rolePolicyDocument)
    {
        var request = new CreateRoleRequest
        {
            RoleName = roleName,
            AssumeRolePolicyDocument = rolePolicyDocument,
        };

        var response = await _IAMService.CreateRoleAsync(request);
        return response.Role.Arn;
    }


    /// <summary>
    /// Create an IAM service-linked role.
    /// </summary>
    /// <param name="serviceName">The name of the AWS Service.</param>
    /// <param name="description">A description of the IAM service-linked role.</
param>
    /// <returns>The IAM role that was created.</returns>
    public async Task<Role> CreateServiceLinkedRoleAsync(string serviceName,
 string description)
    {
        var request = new CreateServiceLinkedRoleRequest
        {
            AWSServiceName = serviceName,
            Description = description
        };

        var response = await _IAMService.CreateServiceLinkedRoleAsync(request);
        return response.Role;
```

```
    }


    /// <summary>
    /// Create an IAM user.
    /// </summary>
    /// <param name="userName">The username for the new IAM user.</param>
    /// <returns>The IAM user that was created.</returns>
    public async Task<User> CreateUserAsync(string userName)
    {
        var response = await _IAMService.CreateUserAsync(new CreateUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// Delete an IAM user's access key.
    /// </summary>
    /// <param name="accessKeyId">The Id for the IAM access key.</param>
    /// <param name="userName">The username of the user that owns the IAM
    /// access key.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteAccessKeyAsync(string accessKeyId, string
userName)
    {
        var response = await _IAMService.DeleteAccessKeyAsync(new
DeleteAccessKeyRequest
        {
            AccessKeyId = accessKeyId,
            UserName = userName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupAsync(string groupName)
    {
```

```
        var response = await _IAMService.DeleteGroupAsync(new DeleteGroupRequest
{ GroupName = groupName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM policy associated with an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group associated with the
    /// policy.</param>
    /// <param name="policyName">The name of the policy to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteGroupPolicyAsync(string groupName, string
policyName)
    {
        var request = new DeleteGroupPolicyRequest()
        {
            GroupName = groupName,
            PolicyName = policyName,
        };

        var response = await _IAMService.DeleteGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM policy.
    /// </summary>
    /// <param name="policyArn">The Amazon Resource Name (ARN) of the policy to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeletePolicyAsync(string policyArn)
    {
        var response = await _IAMService.DeletePolicyAsync(new
DeletePolicyRequest { PolicyArn = policyArn });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role.
    /// </summary>
```

```
    /// <param name="roleName">The name of the IAM role to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRoleAsync(string roleName)
    {
        var response = await _IAMService.DeleteRoleAsync(new DeleteRoleRequest
 { RoleName = roleName });
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM role policy.
    /// </summary>
    /// <param name="roleName">The name of the IAM role.</param>
    /// <param name="policyName">The name of the IAM role policy to delete.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteRolePolicyAsync(string roleName, string
 policyName)
    {
        var response = await _IAMService.DeleteRolePolicyAsync(new
 DeleteRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
        });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an IAM user.
    /// </summary>
    /// <param name="userName">The username of the IAM user to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteUserAsync(string userName)
    {
        var response = await _IAMService.DeleteUserAsync(new DeleteUserRequest
 { UserName = userName });

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

```
   /// <summary>
   /// Delete an IAM user policy.
   /// </summary>
   /// <param name="policyName">The name of the IAM policy to delete.</param>
   /// <param name="userName">The username of the IAM user.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DeleteUserPolicyAsync(string policyName, string
userName)
   {
       var response = await _IAMService.DeleteUserPolicyAsync(new
DeleteUserPolicyRequest { PolicyName = policyName, UserName = userName });

       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }


   /// <summary>
   /// Detach an IAM policy from an IAM role.
   /// </summary>
   /// <param name="policyArn">The Amazon Resource Name (ARN) of the IAM
policy.</param>
   /// <param name="roleName">The name of the IAM role.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> DetachRolePolicyAsync(string policyArn, string
roleName)
   {
       var response = await _IAMService.DetachRolePolicyAsync(new
DetachRolePolicyRequest
       {
           PolicyArn = policyArn,
           RoleName = roleName,
       });

       return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
   }


   /// <summary>
   /// Gets the IAM password policy for an AWS account.
   /// </summary>
   /// <returns>The PasswordPolicy for the AWS account.</returns>
   public async Task<PasswordPolicy> GetAccountPasswordPolicyAsync()
   {
```

```
        var response = await _IAMService.GetAccountPasswordPolicyAsync(new
 GetAccountPasswordPolicyRequest());
        return response.PasswordPolicy;
    }


    /// <summary>
    /// Get information about an IAM policy.
    /// </summary>
    /// <param name="policyArn">The IAM policy to retrieve information for.</
param>
    /// <returns>The IAM policy.</returns>
    public async Task<ManagedPolicy> GetPolicyAsync(string policyArn)
    {

        var response = await _IAMService.GetPolicyAsync(new GetPolicyRequest
 { PolicyArn = policyArn });
        return response.Policy;
    }


    /// <summary>
    /// Get information about an IAM role.
    /// </summary>
    /// <param name="roleName">The name of the IAM role to retrieve information
    /// for.</param>
    /// <returns>The IAM role that was retrieved.</returns>
    public async Task<Role> GetRoleAsync(string roleName)
    {
        var response = await _IAMService.GetRoleAsync(new GetRoleRequest
        {
            RoleName = roleName,
        });

        return response.Role;
    }


    /// <summary>
    /// Get information about an IAM user.
    /// </summary>
    /// <param name="userName">The username of the user.</param>
    /// <returns>An IAM user object.</returns>
    public async Task<User> GetUserAsync(string userName)
```

```
    {
        var response = await _IAMService.GetUserAsync(new GetUserRequest
{ UserName = userName });
        return response.User;
    }


    /// <summary>
    /// List the IAM role policies that are attached to an IAM role.
    /// </summary>
    /// <param name="roleName">The IAM role to list IAM policies for.</param>
    /// <returns>A list of the IAM policies attached to the IAM role.</returns>
    public async Task<List<AttachedPolicyType>>
ListAttachedRolePoliciesAsync(string roleName)
    {
        var attachedPolicies = new List<AttachedPolicyType>();
        var attachedRolePoliciesPaginator =
_IAMService.Paginators.ListAttachedRolePolicies(new
ListAttachedRolePoliciesRequest { RoleName = roleName });

        await foreach (var response in attachedRolePoliciesPaginator.Responses)
        {
            attachedPolicies.AddRange(response.AttachedPolicies);
        }

        return attachedPolicies;
    }


    /// <summary>
    /// List IAM groups.
    /// </summary>
    /// <returns>A list of IAM groups.</returns>
    public async Task<List<Group>> ListGroupsAsync()
    {
        var groupsPaginator = _IAMService.Paginators.ListGroups(new
ListGroupsRequest());
        var groups = new List<Group>();

        await foreach (var response in groupsPaginator.Responses)
        {
            groups.AddRange(response.Groups);
        }
```

```
        return groups;
    }


    /// <summary>
    /// List IAM policies.
    /// </summary>
    /// <returns>A list of the IAM policies.</returns>
    public async Task<List<ManagedPolicy>> ListPoliciesAsync()
    {
        var listPoliciesPaginator = _IAMService.Paginators.ListPolicies(new
 ListPoliciesRequest());
        var policies = new List<ManagedPolicy>();

        await foreach (var response in listPoliciesPaginator.Responses)
        {
            policies.AddRange(response.Policies);
        }

        return policies;
    }


    /// <summary>
    /// List IAM role policies.
    /// </summary>
    /// <param name="roleName">The IAM role for which to list IAM policies.</
param>
    /// <returns>A list of IAM policy names.</returns>
    public async Task<List<string>> ListRolePoliciesAsync(string roleName)
    {
        var listRolePoliciesPaginator =
 _IAMService.Paginators.ListRolePolicies(new ListRolePoliciesRequest { RoleName =
 roleName });
        var policyNames = new List<string>();

        await foreach (var response in listRolePoliciesPaginator.Responses)
        {
            policyNames.AddRange(response.PolicyNames);
        }

        return policyNames;
    }
```

```
    /// <summary>
    /// List IAM roles.
    /// </summary>
    /// <returns>A list of IAM roles.</returns>
    public async Task<List<Role>> ListRolesAsync()
    {
        var listRolesPaginator = _IAMService.Paginators.ListRoles(new
ListRolesRequest());
        var roles = new List<Role>();

        await foreach (var response in listRolesPaginator.Responses)
        {
            roles.AddRange(response.Roles);
        }

        return roles;
    }


    /// <summary>
    /// List SAML authentication providers.
    /// </summary>
    /// <returns>A list of SAML providers.</returns>
    public async Task<List<SAMLProviderListEntry>> ListSAMLProvidersAsync()
    {
        var response = await _IAMService.ListSAMLProvidersAsync(new
ListSAMLProvidersRequest());
        return response.SAMLProviderList;
    }


    /// <summary>
    /// List IAM users.
    /// </summary>
    /// <returns>A list of IAM users.</returns>
    public async Task<List<User>> ListUsersAsync()
    {
        var listUsersPaginator = _IAMService.Paginators.ListUsers(new
ListUsersRequest());
        var users = new List<User>();

        await foreach (var response in listUsersPaginator.Responses)
        {
```

```
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Remove a user from an IAM group.
    /// </summary>
    /// <param name="userName">The username of the user to remove.</param>
    /// <param name="groupName">The name of the IAM group to remove the user
from.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> RemoveUserFromGroupAsync(string userName, string
groupName)
    {
        // Remove the user from the group.
        var removeUserRequest = new RemoveUserFromGroupRequest()
        {
            UserName = userName,
            GroupName = groupName,
        };

        var response = await
_IAMService.RemoveUserFromGroupAsync(removeUserRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM group.
    /// </summary>
    /// <param name="groupName">The name of the IAM group.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutGroupPolicyAsync(string groupName, string
policyName, string policyDocument)
    {
        var request = new PutGroupPolicyRequest
        {
            GroupName = groupName,
```

```
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutGroupPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Update the inline policy document embedded in a role.
    /// </summary>
    /// <param name="policyName">The name of the policy to embed.</param>
    /// <param name="roleName">The name of the role to update.</param>
    /// <param name="policyDocument">The policy document that defines the role.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutRolePolicyAsync(string policyName, string
 roleName, string policyDocument)
    {
        var request = new PutRolePolicyRequest
        {
            PolicyName = policyName,
            RoleName = roleName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutRolePolicyAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Add or update an inline policy document that is embedded in an IAM user.
    /// </summary>
    /// <param name="userName">The name of the IAM user.</param>
    /// <param name="policyName">The name of the IAM policy.</param>
    /// <param name="policyDocument">The policy document defining the IAM
 policy.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> PutUserPolicyAsync(string userName, string
 policyName, string policyDocument)
    {
        var request = new PutUserPolicyRequest
```

```
        {
            UserName = userName,
            PolicyName = policyName,
            PolicyDocument = policyDocument
        };

        var response = await _IAMService.PutUserPolicyAsync(request);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }

    /// <summary>
    /// Wait for a new access key to be ready to use.
    /// </summary>
    /// <param name="accessKeyId">The Id of the access key.</param>
    /// <returns>A boolean value indicating the success of the action.</returns>
    public async Task<bool> WaitUntilAccessKeyIsReady(string accessKeyId)
    {
        var keyReady = false;

        do
        {
            try
            {
                var response = await _IAMService.GetAccessKeyLastUsedAsync(
                    new GetAccessKeyLastUsedRequest { AccessKeyId =
 accessKeyId });
                if (response.UserName is not null)
                {
                    keyReady = true;
                }
            }
            catch (NoSuchEntityException)
            {
                keyReady = false;
            }
        } while (!keyReady);

        return keyReady;
    }
}



using Microsoft.Extensions.Configuration;
```

```
namespace IAMBasics;

public class IAMBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the AWS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
            .AddTransient<IAMWrapper>()
            .AddTransient<UIWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<IAMBasics>();


        IConfiguration configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // Values needed for user, role, and policies.
        string userName = configuration["UserName"]!;
        string s3PolicyName = configuration["S3PolicyName"]!;
        string roleName = configuration["RoleName"]!;


        var iamWrapper = host.Services.GetRequiredService<IAMWrapper>();
        var uiWrapper = host.Services.GetRequiredService<UIWrapper>();
```

```
        uiWrapper.DisplayBasicsOverview();
        uiWrapper.PressEnter();

        // First create a user. By default, the new user has
        // no permissions.
        uiWrapper.DisplayTitle("Create User");
        Console.WriteLine($"Creating a new user with user name: {userName}.");
        var user = await iamWrapper.CreateUserAsync(userName);
        var userArn = user.Arn;

        Console.WriteLine($"Successfully created user: {userName} with ARN:
{userArn}.");
        uiWrapper.WaitABit(15, "Now let's wait for the user to be ready for
use.");

        // Define a role policy document that allows the new user
        // to assume the role.
        string assumeRolePolicyDocument = "{" +
          "\"Version\": \"2012-10-17\"," +
          "\"Statement\": [{" +
              "\"Effect\": \"Allow\"," +
              "\"Principal\": {" +
              $" \"AWS\": \"{userArn}\"" +
              "}," +
              "\"Action\": \"sts:AssumeRole\"" +
          "}]" +
        "}";

        // Permissions to list all buckets.
        string policyDocument = "{" +
            "\"Version\": \"2012-10-17\"," +
            " \"Statement\" : [{" +
                " \"Action\" : [\"s3:ListAllMyBuckets\"]," +
                " \"Effect\" : \"Allow\"," +
                " \"Resource\" : \"*\"" +
            "}]" +
        "}";

        // Create an AccessKey for the user.
        uiWrapper.DisplayTitle("Create access key");
        Console.WriteLine("Now let's create an access key for the new user.");
        var accessKey = await iamWrapper.CreateAccessKeyAsync(userName);

        var accessKeyId = accessKey.AccessKeyId;
```

```
        var secretAccessKey = accessKey.SecretAccessKey;

        Console.WriteLine($"We have created the access key with Access key id:
{accessKeyId}.");

        Console.WriteLine("Now let's wait until the IAM access key is ready to
use.");
        var keyReady = await iamWrapper.WaitUntilAccessKeyIsReady(accessKeyId);

        // Now try listing the Amazon Simple Storage Service (Amazon S3)
        // buckets. This should fail at this point because the user doesn't
        // have permissions to perform this task.
        uiWrapper.DisplayTitle("Try to display Amazon S3 buckets");
        Console.WriteLine("Now let's try to display a list of the user's Amazon
S3 buckets.");
        var s3Client1 = new AmazonS3Client(accessKeyId, secretAccessKey);
        var stsClient1 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        var s3Wrapper = new S3Wrapper(s3Client1, stsClient1);
        var buckets = await s3Wrapper.ListMyBucketsAsync();

        Console.WriteLine(buckets is null
            ? "As expected, the call to list the buckets has returned a null
list."
            : "Something went wrong. This shouldn't have worked.");

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Create IAM role");
        Console.WriteLine($"Creating the role: {roleName}");

        // Creating an IAM role to allow listing the S3 buckets. A role name
        // is not case sensitive and must be unique to the account for which it
        // is created.
        var roleArn = await iamWrapper.CreateRoleAsync(roleName,
assumeRolePolicyDocument);

        uiWrapper.PressEnter();

        // Create a policy with permissions to list S3 buckets.
        uiWrapper.DisplayTitle("Create IAM policy");
        Console.WriteLine($"Creating the policy: {s3PolicyName}");
```

```
        Console.WriteLine("with permissions to list the Amazon S3 buckets for the
account.");
        var policy = await iamWrapper.CreatePolicyAsync(s3PolicyName,
policyDocument);

        // Wait 15 seconds for the IAM policy to be available.
        uiWrapper.WaitABit(15, "Waiting for the policy to be available.");

        // Attach the policy to the role you created earlier.
        uiWrapper.DisplayTitle("Attach new IAM policy");
        Console.WriteLine("Now let's attach the policy to the role.");
        await iamWrapper.AttachRolePolicyAsync(policy.Arn, roleName);

        // Wait 15 seconds for the role to be updated.
        Console.WriteLine();
        uiWrapper.WaitABit(15, "Waiting for the policy to be attached.");

        // Use the AWS Security Token Service (AWS STS) to have the user
        // assume the role we created.
        var stsClient2 = new AmazonSecurityTokenServiceClient(accessKeyId,
secretAccessKey);

        // Wait for the new credentials to become valid.
        uiWrapper.WaitABit(10, "Waiting for the credentials to be valid.");

        var assumedRoleCredentials = await
s3Wrapper.AssumeS3RoleAsync("temporary-session", roleArn);

        // Try again to list the buckets using the client created with
        // the new user's credentials. This time, it should work.
        var s3Client2 = new AmazonS3Client(assumedRoleCredentials);

        s3Wrapper.UpdateClients(s3Client2, stsClient2);

        buckets = await s3Wrapper.ListMyBucketsAsync();

        uiWrapper.DisplayTitle("List Amazon S3 buckets");
        Console.WriteLine("This time we should have buckets to list.");
        if (buckets is not null)
        {
            buckets.ForEach(bucket =>
            {
                Console.WriteLine($"{bucket.BucketName} created:
{bucket.CreationDate}");
```

```
            });
        }

        uiWrapper.PressEnter();

        // Now clean up all the resources used in the example.
        uiWrapper.DisplayTitle("Clean up resources");
        Console.WriteLine("Thank you for watching. The IAM Basics demo is
 complete.");
        Console.WriteLine("Please wait while we clean up the resources we
 created.");

        await iamWrapper.DetachRolePolicyAsync(policy.Arn, roleName);

        await iamWrapper.DeletePolicyAsync(policy.Arn);

        await iamWrapper.DeleteRoleAsync(roleName);

        await iamWrapper.DeleteAccessKeyAsync(accessKeyId, userName);

        await iamWrapper.DeleteUserAsync(userName);

        uiWrapper.PressEnter();

        Console.WriteLine("All done cleaning up our resources. Thank you for your
 patience.");
    }
}


namespace IamScenariosCommon;

using System.Net;

/// <summary>
/// A class to perform Amazon Simple Storage Service (Amazon S3) actions for
/// the IAM Basics scenario.
/// </summary>
public class S3Wrapper
{
    private IAmazonS3 _s3Service;
    private IAmazonSecurityTokenService _stsService;

    /// <summary>
```

```
    /// Constructor for the S3Wrapper class.
    /// </summary>
    /// <param name="s3Service">An Amazon S3 client object.</param>
    /// <param name="stsService">An AWS Security Token Service (AWS STS)
    /// client object.</param>
    public S3Wrapper(IAmazonS3 s3Service, IAmazonSecurityTokenService stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }

    /// <summary>
    /// Assumes an AWS Identity and Access Management (IAM) role that allows
    /// Amazon S3 access for the current session.
    /// </summary>
    /// <param name="roleSession">A string representing the current session.</
param>
    /// <param name="roleToAssume">The name of the IAM role to assume.</param>
    /// <returns>Credentials for the newly assumed IAM role.</returns>
    public async Task<Credentials> AssumeS3RoleAsync(string roleSession, string
 roleToAssume)
    {
        // Create the request to use with the AssumeRoleAsync call.
        var request = new AssumeRoleRequest()
        {
            RoleSessionName = roleSession,
            RoleArn = roleToAssume,
        };

        var response = await _stsService.AssumeRoleAsync(request);

        return response.Credentials;
    }


    /// <summary>
    /// Delete an S3 bucket.
    /// </summary>
    /// <param name="bucketName">Name of the S3 bucket to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteBucketAsync(string bucketName)
    {
        var result = await _s3Service.DeleteBucketAsync(new DeleteBucketRequest
 { BucketName = bucketName });
```

```
        return result.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the buckets that are owned by the user's account.
    /// </summary>
    /// <returns>Async Task.</returns>
    public async Task<List<S3Bucket>?> ListMyBucketsAsync()
    {
        try
        {
            // Get the list of buckets accessible by the new user.
            var response = await _s3Service.ListBucketsAsync();

            return response.Buckets;
        }
        catch (AmazonS3Exception ex)
        {
            // Something else went wrong. Display the error message.
            Console.WriteLine($"Error: {ex.Message}");
            return null;
        }
    }

    /// <summary>
    /// Create a new S3 bucket.
    /// </summary>
    /// <param name="bucketName">The name for the new bucket.</param>
    /// <returns>A Boolean value indicating whether the action completed
    /// successfully.</returns>
    public async Task<bool> PutBucketAsync(string bucketName)
    {
        var response = await _s3Service.PutBucketAsync(new PutBucketRequest
{ BucketName = bucketName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Update the client objects with new client objects. This is available
    /// because the scenario uses the methods of this class without and then
    /// with the proper permissions to list S3 buckets.
    /// </summary>
    /// <param name="s3Service">The Amazon S3 client object.</param>
    /// <param name="stsService">The AWS STS client object.</param>
```

```
    public void UpdateClients(IAmazonS3 s3Service, IAmazonSecurityTokenService
 stsService)
    {
        _s3Service = s3Service;
        _stsService = stsService;
    }
}


namespace IamScenariosCommon;

public class UIWrapper
{
    public readonly string SepBar = new('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the IAM Groups scenario.
    /// </summary>
    public void DisplayGroupsOverview()
    {
        Console.Clear();

        DisplayTitle("Welcome to the IAM Groups Demo");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates an Amazon Identity and Access Management
 (IAM) group.");
        Console.WriteLine("\t2. Adds an IAM policy to the IAM group giving it
 full access to Amazon S3.");
        Console.WriteLine("\t3. Creates a new IAM user.");
        Console.WriteLine("\t4. Creates an IAM access key for the user.");
        Console.WriteLine("\t5. Adds the user to the IAM group.");
        Console.WriteLine("\t6. Lists the buckets on the account.");
        Console.WriteLine("\t7. Proves that the user has full Amazon S3 access by
 creating a bucket.");
        Console.WriteLine("\t8. List the buckets again to show the new bucket.");
        Console.WriteLine("\t9. Cleans up all the resources created.");
    }

    /// <summary>
    /// Show information about the IAM Basics scenario.
    /// </summary>
    public void DisplayBasicsOverview()
    {
        Console.Clear();
```

```
        DisplayTitle("Welcome to IAM Basics");
        Console.WriteLine("This example application does the following:");
        Console.WriteLine("\t1. Creates a user with no permissions.");
        Console.WriteLine("\t2. Creates a role and policy that grant
s3:ListAllMyBuckets permission.");
        Console.WriteLine("\t3. Grants the user permission to assume the role.");
        Console.WriteLine("\t4. Creates an S3 client object as the user and tries
to list buckets (this will fail).");
        Console.WriteLine("\t5. Gets temporary credentials by assuming the
role.");
        Console.WriteLine("\t6. Creates a new S3 client object with the temporary
credentials and lists the buckets (this will succeed).");
        Console.WriteLine("\t7. Deletes all the resources.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPress <Enter> to continue. ");
        _ = Console.ReadLine();
        Console.WriteLine();
    }

    /// <summary>
    /// Pad a string with spaces to center it on the console display.
    /// </summary>
    /// <param name="strToCenter">The string to be centered.</param>
    /// <returns>The padded string.</returns>
    public string CenterString(string strToCenter)
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title, and another
    /// line of hyphens.
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
```

```
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }

    /// <summary>
    /// Display a countdown and wait for a number of seconds.
    /// </summary>
    /// <param name="numSeconds">The number of seconds to wait.</param>
    public void WaitABit(int numSeconds, string msg)
    {
        Console.WriteLine(msg);

        // Wait for the requested number of seconds.
        for (int i = numSeconds; i > 0; i--)
        {
            System.Threading.Thread.Sleep(1000);
            Console.Write($"{i}...");
        }

        PressEnter();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica

- DetachRolePolitica
- PutUserPolitica

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
###############################################################################
# function iam_create_user_assume_role
#
# Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
#
#     "IAM access" permissions are needed to run this code.
#     "STS assume role" permissions are needed to run this code. (Note: It might
 be necessary to
#           create a custom policy).
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function iam_create_user_assume_role() {
  {
    if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

      source ./iam_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the IAM create user and assume role demo."
  echo
  echo "This demo will create an IAM user, create an IAM role, and apply the role
 to the user."
```

```
  echo_repeat "*" 88
  echo

  echo -n "Enter a name for a new IAM user: "
  get_input
  user_name=$get_input_result

  local user_arn
  user_arn=$(iam_create_user -u "$user_name")

  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created demo IAM user named $user_name"
  else
    errecho "$user_arn"
    errecho "The user failed to create. This demo will exit."
    return 1
  fi

  local access_key_response
  access_key_response=$(iam_create_user_access_key -u "$user_name")
  # shellcheck disable=SC2181
  if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
  fi

  IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
  local key_name=${access_key_values[0]}
  local key_secret=${access_key_values[1]}

  echo "Created access key named $key_name"

  echo "Wait 10 seconds for the user to be ready."
  sleep 10
  echo_repeat "*" 88
  echo

  local iam_role_name
  iam_role_name=$(generate_random_name "test-role")
  echo "Creating a role named $iam_role_name with user $user_name as the
principal."
```

```
  local assume_role_policy_document="{
    \"Version\": \"2012-10-17\",
    \"Statement\": [{
        \"Effect\": \"Allow\",
        \"Principal\": {\"AWS\": \"$user_arn\"},
        \"Action\": \"sts:AssumeRole\"
        }]
    }"

  local role_arn
  role_arn=$(iam_create_role -n "$iam_role_name" -p
"$assume_role_policy_document")

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created IAM role named $iam_role_name"
  else
    errecho "The role failed to create. This demo will exit."
    clean_up "$user_name" "$key_name"
    return 1
  fi

  local policy_name
  policy_name=$(generate_random_name "test-policy")
  local policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]}"

  local policy_arn
  policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
  # shellcheck disable=SC2181
  if [[ ${?} == 0 ]]; then
    echo "Created  IAM policy named $policy_name"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
  fi

  if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
```

```
  else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
  fi

  local assume_role_policy_document="{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"$role_arn\"}]}"

  local assume_role_policy_name
  assume_role_policy_name=$(generate_random_name "test-assume-role-")

  # shellcheck disable=SC2181
  local assume_role_policy_arn
  assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    echo "Created  IAM policy named $assume_role_policy_name for sts assume role"
  else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn"
    return 1
  fi

  echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
  sleep 10
  echo_repeat "*" 88
  echo

  echo "Try to list buckets without the new user assuming the role."
  echo_repeat "*" 88
  echo

  # Set the environment variables for the created user.
  # bashsupport disable=BP2001
  export AWS_ACCESS_KEY_ID=$key_name
  # bashsupport disable=BP2001
```

```
 export AWS_SECRET_ACCESS_KEY=$key_secret

 local buckets
 buckets=$(s3_list_buckets)

 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   local bucket_count
   bucket_count=$(echo "$buckets" | wc -w | xargs)
   echo "There are $bucket_count buckets in the account. This should not have
happened."
 else
   errecho "Because the role with permissions has not been assumed, listing
buckets failed."
 fi

 echo
 echo_repeat "*" 88
 echo "Now assume the role $iam_role_name and list the buckets."
 echo_repeat "*" 88
 echo

 local credentials

 credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
 # shellcheck disable=SC2181
 if [ ${?} == 0 ]; then
   echo "Assumed role $iam_role_name"
 else
   errecho "Failed to assume role."
   export AWS_ACCESS_KEY_ID=""
   export AWS_SECRET_ACCESS_KEY=""
   clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
"$policy_arn" "$assume_role_policy_arn"
   return 1
 fi

 IFS=$'\t ' read -r -a credentials <<<"$credentials"

 export AWS_ACCESS_KEY_ID=${credentials[0]}
 export AWS_SECRET_ACCESS_KEY=${credentials[1]}
 # bashsupport disable=BP2001
 export AWS_SESSION_TOKEN=${credentials[2]}
```

```
  buckets=$(s3_list_buckets)

  # shellcheck disable=SC2181
  if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets
 succeeded because of "
    echo "the assumed role."
  else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
 "$policy_arn" "$assume_role_policy_arn"
    return 1
  fi

  local result=0
  export AWS_ACCESS_KEY_ID=""
  export AWS_SECRET_ACCESS_KEY=""

  echo
  echo_repeat "*" 88
  echo "The created resources will now be deleted."
  echo_repeat "*" 88
  echo

  clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
 "$assume_role_policy_arn"

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    result=1
  fi

  return $result
}
```

Le funzioni IAM utilizzate in questo scenario.

```
########################################################################
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
 (IAM) user already exists.
#
# Parameters:
#       $1 - The name of the IAM user to check.
#
# Returns:
#       0 - If the user already exists.
#       1 - If the user doesn't exist.
########################################################################
function iam_user_exists() {
  local user_name
  user_name=$1

  # Check whether the IAM user already exists.
  # We suppress all output - we're interested only in the return code.

  local errors
  errors=$(aws iam get-user \
    --user-name "$user_name" 2>&1 >/dev/null)

  local error_code=${?}

  if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
  else
    if [[ $errors != *"error"*"(NoSuchEntity)"* ]]; then
      aws_cli_error_log $error_code
      errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
  fi
}

########################################################################
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
```

```
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       The ARN of the user.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user. It must be unique within the
 account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
```

```
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user already exists, we don't want to try to create it.
  if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
  fi

  response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#       -u user_name -- The name of the IAM user.
#       [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#       [access_key_id access_key_secret]
#     And:
#       0 - If successful.
#       1 - If it fails.
```

```
########################################################################################
function iam_create_user_access_key() {
  local user_name file_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_json -- The assume role policy document.
#
# Returns:
#       The ARN of the role.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_role() {
  local role_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
```

```
  echo "  -n role_name   The name of the IAM role."
  echo "  -p policy_json -- The assume role policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-role \
  --role-name "$role_name" \
  --assume-role-policy-document "$policy_document" \
  --output text \
  --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#       -n policy_name -- The name of the IAM policy.
#       -p policy_json -- The policy document.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_create_policy() {
  local policy_name policy_document response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
```

```
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_attach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
 role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
```

```
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#       -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_detach_role_policy() {
  local role_name policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
```

```
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo "  -n role_name   The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      p) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

  local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}

###############################################################################
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#       -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_policy() {
  local policy_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
```

```
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an WS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
```

```
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user.
#       -k access_key -- The access key to delete.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_access_key() {
  local user_name access_key response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an WS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
```

```
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
```

```
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#       -u user_name  -- The name of the user to create.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must
 supply a username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento dei comandi AWS CLI .
  - AttachRolePolitica

- CreateAccessChiave
- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessChiave
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

C++

SDK per C++

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace AwsDoc {
    namespace IAM {

        //! Cleanup by deleting created entities.
        /*!
          \sa DeleteCreatedEntities
          \param client: IAM client.
          \param role: IAM role.
          \param user: IAM user.
          \param policy: IAM policy.
        */
        static bool DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                          const Aws::IAM::Model::Role &role,
                                          const Aws::IAM::Model::User &user,
                                          const Aws::IAM::Model::Policy &policy);
```

```
    }

    static const int LIST_BUCKETS_WAIT_SEC = 20;

    static const char ALLOCATION_TAG[] = "example_code";
}

//! Scenario to create an IAM user, create an IAM role, and apply the role to the
 user.
// "IAM access" permissions are needed to run this code.
// "STS assume role" permissions are needed to run this code. (Note: It might be
 necessary to
//    create a custom policy).
/*!
  \sa iamCreateUserAssumeRoleScenario
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
*/
bool AwsDoc::IAM::iamCreateUserAssumeRoleScenario(
        const Aws::Client::ClientConfiguration &clientConfig) {

    Aws::IAM::IAMClient client(clientConfig);
    Aws::IAM::Model::User user;
    Aws::IAM::Model::Role role;
    Aws::IAM::Model::Policy policy;

    // 1. Create a user.
    {
        Aws::IAM::Model::CreateUserRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String userName = "iam-demo-user-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetUserName(userName);

        Aws::IAM::Model::CreateUserOutcome outcome = client.CreateUser(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Error creating IAM user " << userName << ":" <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }
        else {
            std::cout << "Successfully created IAM user " << userName <<
 std::endl;
        }
```

```
        user = outcome.GetResult().GetUser();
    }

    // 2. Create a role.
    {
        // Get the IAM user for the current client in order to access its ARN.
        Aws::String iamUserArn;
        {
            Aws::IAM::Model::GetUserRequest request;
            Aws::IAM::Model::GetUserOutcome outcome = client.GetUser(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error getting Iam user. " <<
                          outcome.GetError().GetMessage() << std::endl;

                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }
            else {
                std::cout << "Successfully retrieved Iam user "
                          << outcome.GetResult().GetUser().GetUserName()
                          << std::endl;
            }

            iamUserArn = outcome.GetResult().GetUser().GetArn();
        }

        Aws::IAM::Model::CreateRoleRequest request;

        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleName = "iam-demo-role-" +
                               Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleName(roleName);

        // Build policy document for role.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");

        Aws::Utils::Document jsonPrincipal;
        jsonPrincipal.WithString("AWS", iamUserArn);
        jsonStatement.WithObject("Principal", jsonPrincipal);
        jsonStatement.WithString("Action", "sts:AssumeRole");
        jsonStatement.WithObject("Condition", Aws::Utils::Document());
```

```
        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Setting policy for role\n   "
                  << policyDocument.View().WriteCompact() << std::endl;

        // Set role policy document as JSON string.

request.SetAssumeRolePolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreateRoleOutcome outcome = client.CreateRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating role. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a role with name " << roleName
                      << std::endl;
        }

        role = outcome.GetResult().GetRole();
    }

    // 3. Create an IAM policy.
    {
        Aws::IAM::Model::CreatePolicyRequest request;
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String policyName = "iam-demo-policy-" +
                                 Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetPolicyName(policyName);

        // Build IAM policy document.
        Aws::Utils::Document jsonStatement;
        jsonStatement.WithString("Effect", "Allow");
        jsonStatement.WithString("Action", "s3:ListAllMyBuckets");
        jsonStatement.WithString("Resource", "arn:aws:s3:::*");
```

```
        Aws::Utils::Document policyDocument;
        policyDocument.WithString("Version", "2012-10-17");

        Aws::Utils::Array<Aws::Utils::Document> statements(1);
        statements[0] = jsonStatement;
        policyDocument.WithArray("Statement", statements);

        std::cout << "Creating a policy.\n   " <<
policyDocument.View().WriteCompact()
                  << std::endl;

        // Set IAM policy document as JSON string.
        request.SetPolicyDocument(policyDocument.View().WriteCompact());

        Aws::IAM::Model::CreatePolicyOutcome outcome =
client.CreatePolicy(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error creating policy. " <<
                      outcome.GetError().GetMessage() << std::endl;

            DeleteCreatedEntities(client, role, user, policy);
            return false;
        }
        else {
            std::cout << "Successfully created a policy with name, " <<
policyName <<
                      "." << std::endl;
        }

        policy = outcome.GetResult().GetPolicy();
    }

    // 4. Assume the new role using the AWS Security Token Service (STS).
    Aws::STS::Model::Credentials credentials;
    {
        Aws::STS::STSClient stsClient(clientConfig);

        Aws::STS::Model::AssumeRoleRequest request;
        request.SetRoleArn(role.GetArn());
        Aws::String uuid = Aws::Utils::UUID::RandomUUID();
        Aws::String roleSessionName = "iam-demo-role-session-" +

Aws::Utils::StringUtils::ToLower(uuid.c_str());
        request.SetRoleSessionName(roleSessionName);
```

```
        Aws::STS::Model::AssumeRoleOutcome assumeRoleOutcome;

        // Repeatedly call AssumeRole, because there is often a delay
        // before the role is available to be assumed.
        // Repeat at most 20 times when access is denied.
        int count = 0;
        while (true) {
            assumeRoleOutcome = stsClient.AssumeRole(request);
            if (!assumeRoleOutcome.IsSuccess()) {
                if (count > 20 ||
                    assumeRoleOutcome.GetError().GetErrorType() !=
                    Aws::STS::STSErrors::ACCESS_DENIED) {
                    std::cerr << "Error assuming role after 20 tries. " <<
                              assumeRoleOutcome.GetError().GetMessage() <<
std::endl;

                    DeleteCreatedEntities(client, role, user, policy);
                    return false;
                }
                std::this_thread::sleep_for(std::chrono::seconds(1));
            }
            else {
                std::cout << "Successfully assumed the role after " << count
                          << " seconds." << std::endl;
                break;
            }
            count++;
        }

        credentials = assumeRoleOutcome.GetResult().GetCredentials();
    }


    // 5. List objects in the bucket (This should fail).
    {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
s3Client.ListBuckets();
```

```
       if (!listBucketsOutcome.IsSuccess()) {
           if (listBucketsOutcome.GetError().GetErrorType() !=
               Aws::S3::S3Errors::ACCESS_DENIED) {
               std::cerr << "Could not lists buckets. " <<
                         listBucketsOutcome.GetError().GetMessage() <<
std::endl;
           }
           else {
               std::cout
                       << "Access to list buckets denied because privileges have
not been applied."
                       << std::endl;
           }
       }
       else {
           std::cerr
                   << "Successfully retrieved bucket lists when this should not
happen."
                   << std::endl;
       }
   }

   // 6. Attach the policy to the role.
   {
       Aws::IAM::Model::AttachRolePolicyRequest request;
       request.SetRoleName(role.GetRoleName());
       request.WithPolicyArn(policy.GetArn());

       Aws::IAM::Model::AttachRolePolicyOutcome outcome =
client.AttachRolePolicy(
               request);
       if (!outcome.IsSuccess()) {
           std::cerr << "Error creating policy. " <<
                     outcome.GetError().GetMessage() << std::endl;

           DeleteCreatedEntities(client, role, user, policy);
           return false;
       }
       else {
           std::cout << "Successfully attached the policy with name, "
                     << policy.GetPolicyName() <<
                     ", to the role, " << role.GetRoleName() << "." <<
std::endl;
       }
```

```
    }

    int count = 0;
    // 7. List objects in the bucket (this should succeed).
    // Repeatedly call ListBuckets, because there is often a delay
    // before the policy with ListBucket permissions has been applied to the
 role.
    // Repeat at most LIST_BUCKETS_WAIT_SEC times when access is denied.
    while (true) {
        Aws::S3::S3Client s3Client(
                Aws::Auth::AWSCredentials(credentials.GetAccessKeyId(),
                                          credentials.GetSecretAccessKey(),
                                          credentials.GetSessionToken()),
                Aws::MakeShared<Aws::S3::S3EndpointProvider>(ALLOCATION_TAG),
                clientConfig);
        Aws::S3::Model::ListBucketsOutcome listBucketsOutcome =
 s3Client.ListBuckets();
        if (!listBucketsOutcome.IsSuccess()) {
            if ((count > LIST_BUCKETS_WAIT_SEC) ||
                listBucketsOutcome.GetError().GetErrorType() !=
                Aws::S3::S3Errors::ACCESS_DENIED) {
                std::cerr << "Could not lists buckets after " <<
 LIST_BUCKETS_WAIT_SEC << " seconds. " <<
                          listBucketsOutcome.GetError().GetMessage() <<
 std::endl;
                DeleteCreatedEntities(client, role, user, policy);
                return false;
            }

            std::this_thread::sleep_for(std::chrono::seconds(1));
        }
        else {

            std::cout << "Successfully retrieved bucket lists after " << count
                      << " seconds." << std::endl;
            break;
        }
        count++;
    }

    // 8. Delete all the created resources.
    return DeleteCreatedEntities(client, role, user, policy);
}
```

```
bool AwsDoc::IAM::DeleteCreatedEntities(const Aws::IAM::IAMClient &client,
                                        const Aws::IAM::Model::Role &role,
                                        const Aws::IAM::Model::User &user,
                                        const Aws::IAM::Model::Policy &policy) {
    bool result = true;
    if (policy.ArnHasBeenSet()) {
        // Detach the policy from the role.
        {
            Aws::IAM::Model::DetachRolePolicyRequest request;
            request.SetPolicyArn(policy.GetArn());
            request.SetRoleName(role.GetRoleName());

            Aws::IAM::Model::DetachRolePolicyOutcome outcome =
 client.DetachRolePolicy(
                    request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error Detaching policy from roles. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully detached the policy with arn "
                          << policy.GetArn()
                          << " from role " << role.GetRoleName() << "." <<
 std::endl;
            }
        }

        // Delete the policy.
        {
            Aws::IAM::Model::DeletePolicyRequest request;
            request.WithPolicyArn(policy.GetArn());

            Aws::IAM::Model::DeletePolicyOutcome outcome =
 client.DeletePolicy(request);
            if (!outcome.IsSuccess()) {
                std::cerr << "Error deleting policy. " <<
                          outcome.GetError().GetMessage() << std::endl;
                result = false;
            }
            else {
                std::cout << "Successfully deleted the policy with arn "
                          << policy.GetArn() << std::endl;
            }
```

```
        }

    }

    if (role.RoleIdHasBeenSet()) {
        // Delete the role.
        Aws::IAM::Model::DeleteRoleRequest request;
        request.SetRoleName(role.GetRoleName());

        Aws::IAM::Model::DeleteRoleOutcome outcome = client.DeleteRole(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting role. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the role with name "
                      << role.GetRoleName() << std::endl;
        }
    }

    if (user.ArnHasBeenSet()) {
        // Delete the user.
        Aws::IAM::Model::DeleteUserRequest request;
        request.WithUserName(user.GetUserName());

        Aws::IAM::Model::DeleteUserOutcome outcome = client.DeleteUser(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error deleting user. " <<
                      outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
        else {
            std::cout << "Successfully deleted the user with name "
                      << user.GetUserName() << std::endl;
        }
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for C++ .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
// AssumeRoleScenario shows you how to use the AWS Identity and Access Management
 (IAM)
// service to perform the following actions:
//
// 1. Create a user who has no permissions.
// 2. Create a role that grants permission to list Amazon Simple Storage Service
//    (Amazon S3) buckets for the account.
// 3. Add a policy to let the user assume the role.
```

```
// 4. Try and fail to list buckets without permissions.
// 5. Assume the role and list S3 buckets using temporary credentials.
// 6. Delete the policy, role, and user.
type AssumeRoleScenario struct {
 sdkConfig aws.Config
 accountWrapper actions.AccountWrapper
 policyWrapper actions.PolicyWrapper
 roleWrapper actions.RoleWrapper
 userWrapper actions.UserWrapper
 questioner demotools.IQuestioner
 helper IScenarioHelper
 isTestRun bool
}

// NewAssumeRoleScenario constructs an AssumeRoleScenario instance from a
 configuration.
// It uses the specified config to get an IAM client and create wrappers for the
 actions
// used in the scenario.
func NewAssumeRoleScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
  helper IScenarioHelper) AssumeRoleScenario {
 iamClient := iam.NewFromConfig(sdkConfig)
 return AssumeRoleScenario{
  sdkConfig:   sdkConfig,
  accountWrapper: actions.AccountWrapper{IamClient: iamClient},
  policyWrapper:  actions.PolicyWrapper{IamClient: iamClient},
  roleWrapper:    actions.RoleWrapper{IamClient: iamClient},
  userWrapper:    actions.UserWrapper{IamClient: iamClient},
  questioner:     questioner,
  helper:         helper,
 }
}

// addTestOptions appends the API options specified in the original configuration
 to
// another configuration. This is used to attach the middleware stubber to
 clients
// that are constructed during the scenario, which is needed for unit testing.
func (scenario AssumeRoleScenario) addTestOptions(scenarioConfig *aws.Config) {
 if scenario.isTestRun {
  scenarioConfig.APIOptions = append(scenarioConfig.APIOptions,
 scenario.sdkConfig.APIOptions...)
 }
```

```
}

// Run runs the interactive scenario.
func (scenario AssumeRoleScenario) Run() {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong with the demo.\n")
   log.Println(r)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the AWS Identity and Access Management (IAM) assume role
 demo.")
 log.Println(strings.Repeat("-", 88))

 user := scenario.CreateUser()
 accessKey := scenario.CreateAccessKey(user)
 role := scenario.CreateRoleAndPolicies(user)
 noPermsConfig := scenario.ListBucketsWithoutPermissions(accessKey)
 scenario.ListBucketsWithAssumedRole(noPermsConfig, role)
 scenario.Cleanup(user, role)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateUser creates a new IAM user. This user has no permissions.
func (scenario AssumeRoleScenario) CreateUser() *types.User {
 log.Println("Let's create an example user with no permissions.")
 userName := scenario.questioner.Ask("Enter a name for the example user:",
 demotools.NotEmpty{})
 user, err := scenario.userWrapper.GetUser(userName)
 if err != nil {
  panic(err)
 }
 if user == nil {
  user, err = scenario.userWrapper.CreateUser(userName)
  if err != nil {
   panic(err)
  }
  log.Printf("Created user %v.\n", *user.UserName)
 } else {
```

```
  log.Printf("User %v already exists.\n", *user.UserName)
 }
 log.Println(strings.Repeat("-", 88))
 return user
}

// CreateAccessKey creates an access key for the user.
func (scenario AssumeRoleScenario) CreateAccessKey(user *types.User)
 *types.AccessKey {
 accessKey, err := scenario.userWrapper.CreateAccessKeyPair(*user.UserName)
 if err != nil {
  panic(err)
 }
 log.Printf("Created access key %v for your user.", *accessKey.AccessKeyId)
 log.Println("Waiting a few seconds for your user to be ready...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return accessKey
}

// CreateRoleAndPolicies creates a policy that grants permission to list S3
 buckets for
// the current account and attaches the policy to a newly created role. It also
 adds an
// inline policy to the specified user that grants the user permission to assume
 the role.
func (scenario AssumeRoleScenario) CreateRoleAndPolicies(user *types.User)
 *types.Role {
 log.Println("Let's create a role and policy that grant permission to list S3
 buckets.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 listBucketsRole, err :=
 scenario.roleWrapper.CreateRole(scenario.helper.GetName(), *user.Arn)
 if err != nil {panic(err)}
 log.Printf("Created role %v.\n", *listBucketsRole.RoleName)
 listBucketsPolicy, err := scenario.policyWrapper.CreatePolicy(
  scenario.helper.GetName(), []string{"s3:ListAllMyBuckets"}, "arn:aws:s3:::*")
 if err != nil {panic(err)}
 log.Printf("Created policy %v.\n", *listBucketsPolicy.PolicyName)
 err = scenario.roleWrapper.AttachRolePolicy(*listBucketsPolicy.Arn,
 *listBucketsRole.RoleName)
 if err != nil {panic(err)}
 log.Printf("Attached policy %v to role %v.\n", *listBucketsPolicy.PolicyName,
  *listBucketsRole.RoleName)
```

```
 err = scenario.userWrapper.CreateUserPolicy(*user.UserName,
 scenario.helper.GetName(),
  []string{"sts:AssumeRole"}, *listBucketsRole.Arn)
 if err != nil {panic(err)}
 log.Printf("Created an inline policy for user %v that lets the user assume the
 role.\n",
  *user.UserName)
 log.Println("Let's give AWS a few seconds to propagate these new resources and
 connections...")
 scenario.helper.Pause(10)
 log.Println(strings.Repeat("-", 88))
 return listBucketsRole
}

// ListBucketsWithoutPermissions creates an Amazon S3 client from the user's
 access key
// credentials and tries to list buckets for the account. Because the user does
 not have
// permission to perform this action, the action fails.
func (scenario AssumeRoleScenario) ListBucketsWithoutPermissions(accessKey
 *types.AccessKey) *aws.Config {
  log.Println("Let's try to list buckets without permissions. This should return
 an AccessDenied error.")
  scenario.questioner.Ask("Press Enter when you're ready.")
  noPermsConfig, err := config.LoadDefaultConfig(context.TODO(),
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *accessKey.AccessKeyId, *accessKey.SecretAccessKey, ""),
 ))
  if err != nil {panic(err)}

  // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&noPermsConfig)

  s3Client := s3.NewFromConfig(noPermsConfig)
  _, err = s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
  if err != nil {
   // The SDK for Go does not model the AccessDenied error, so check ErrorCode
 directly.
  var ae smithy.APIError
  if errors.As(err, &ae) {
   switch ae.ErrorCode() {
   case "AccessDenied":
```

```
    log.Println("Got AccessDenied error, which is the expected result because\n"
 +
     "the ListBuckets call was made without permissions.")
   default:
    log.Println("Expected AccessDenied, got something else.")
    panic(err)
   }
  }
  } else {
   log.Println("Expected AccessDenied error when calling ListBuckets without
 permissions,\n" +
    "but the call succeeded. Continuing the example anyway...")
 }
 log.Println(strings.Repeat("-", 88))
  return &noPermsConfig
}

// ListBucketsWithAssumedRole performs the following actions:
//
// 1. Creates an AWS Security Token Service (AWS STS) client from the config
 created from
//   the user's access key credentials.
// 2. Gets temporary credentials by assuming the role that grants permission to
 list the
//    buckets.
// 3. Creates an Amazon S3 client from the temporary credentials.
// 4. Lists buckets for the account. Because the temporary credentials are
 generated by
//    assuming the role that grants permission, the action succeeds.
func (scenario AssumeRoleScenario) ListBucketsWithAssumedRole(noPermsConfig
 *aws.Config, role *types.Role) {
 log.Println("Let's assume the role that grants permission to list buckets and
 try again.")
 scenario.questioner.Ask("Press Enter when you're ready.")
 stsClient := sts.NewFromConfig(*noPermsConfig)
 tempCredentials, err := stsClient.AssumeRole(context.TODO(),
 &sts.AssumeRoleInput{
  RoleArn:           role.Arn,
  RoleSessionName:   aws.String("AssumeRoleExampleSession"),
  DurationSeconds:   aws.Int32(900),
 })
 if err != nil {
  log.Printf("Couldn't assume role %v.\n", *role.RoleName)
  panic(err)
```

```
 }
 log.Printf("Assumed role %v, got temporary credentials.\n", *role.RoleName)
 assumeRoleConfig, err := config.LoadDefaultConfig(context.TODO(),
  config.WithCredentialsProvider(credentials.NewStaticCredentialsProvider(
   *tempCredentials.Credentials.AccessKeyId,
   *tempCredentials.Credentials.SecretAccessKey,
   *tempCredentials.Credentials.SessionToken),
  ),
 )
 if err != nil {panic(err)}

 // Add test options if this is a test run. This is needed only for testing
 purposes.
 scenario.addTestOptions(&assumeRoleConfig)

 s3Client := s3.NewFromConfig(assumeRoleConfig)
 result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
 if err != nil {
  log.Println("Couldn't list buckets with assumed role credentials.")
  panic(err)
 }
 log.Println("Successfully called ListBuckets with assumed role credentials, \n"
 +
  "here are some of them:")
 for i := 0; i < len(result.Buckets) && i < 5; i++ {
  log.Printf("\t%v\n", *result.Buckets[i].Name)
 }
 log.Println(strings.Repeat("-", 88))
}

// Cleanup deletes all resources created for the scenario.
func (scenario AssumeRoleScenario) Cleanup(user *types.User, role *types.Role) {
 if scenario.questioner.AskBool(
  "Do you want to delete the resources created for this example? (y/n)", "y",
 ) {
   policies, err := scenario.roleWrapper.ListAttachedRolePolicies(*role.RoleName)
   if err != nil {panic(err)}
   for _, policy := range policies {
    err = scenario.roleWrapper.DetachRolePolicy(*role.RoleName,
 *policy.PolicyArn)
    if err != nil {panic(err)}
    err = scenario.policyWrapper.DeletePolicy(*policy.PolicyArn)
    if err != nil {panic(err)}
    log.Printf("Detached policy %v from role %v and deleted the policy.\n",
```

```
      *policy.PolicyName, *role.RoleName)
    }
    err = scenario.roleWrapper.DeleteRole(*role.RoleName)
    if err != nil {panic(err)}
    log.Printf("Deleted role %v.\n", *role.RoleName)

    userPols, err := scenario.userWrapper.ListUserPolicies(*user.UserName)
    if err != nil {panic(err)}
    for _, userPol := range userPols {
     err = scenario.userWrapper.DeleteUserPolicy(*user.UserName, userPol)
     if err != nil {panic(err)}
     log.Printf("Deleted policy %v from user %v.\n", userPol, *user.UserName)
    }
    keys, err := scenario.userWrapper.ListAccessKeys(*user.UserName)
    if err != nil {panic(err)}
    for _, key := range keys {
     err = scenario.userWrapper.DeleteAccessKey(*user.UserName, *key.AccessKeyId)
     if err != nil {panic(err)}
     log.Printf("Deleted access key %v from user %v.\n", *key.AccessKeyId,
 *user.UserName)
    }
    err = scenario.userWrapper.DeleteUser(*user.UserName)
    if err != nil {panic(err)}
    log.Printf("Deleted user %v.\n", *user.UserName)
    log.Println(strings.Repeat("-", 88))
 }

}
```

Definisci una struttura che racchiude le azioni dell'account.

```
// AccountWrapper encapsulates AWS Identity and Access Management (IAM) account
 actions
// used in the examples.
// It contains an IAM service client that is used to perform account actions.
type AccountWrapper struct {
 IamClient *iam.Client
}
```

```
// GetAccountPasswordPolicy gets the account password policy for the current
 account.
// If no policy has been set, a NoSuchEntityException is error is returned.
func (wrapper AccountWrapper) GetAccountPasswordPolicy() (*types.PasswordPolicy,
 error) {
 var pwPolicy *types.PasswordPolicy
 result, err := wrapper.IamClient.GetAccountPasswordPolicy(context.TODO(),
  &iam.GetAccountPasswordPolicyInput{})
 if err != nil {
  log.Printf("Couldn't get account password policy. Here's why: %v\n", err)
 } else {
  pwPolicy = result.PasswordPolicy
 }
 return pwPolicy, err
}



// ListSAMLProviders gets the SAML providers for the account.
func (wrapper AccountWrapper) ListSAMLProviders() ([]types.SAMLProviderListEntry,
 error) {
 var providers []types.SAMLProviderListEntry
 result, err := wrapper.IamClient.ListSAMLProviders(context.TODO(),
 &iam.ListSAMLProvidersInput{})
 if err != nil {
  log.Printf("Couldn't list SAML providers. Here's why: %v\n", err)
 } else {
  providers = result.SAMLProviderList
 }
 return providers, err
}
```

Definisci una struttura che racchiude le azioni della policy.

```
// PolicyDocument defines a policy document as a Go struct that can be serialized
// to JSON.
type PolicyDocument struct {
 Version string
 Statement []PolicyStatement
```

```
}

// PolicyStatement defines a statement in a policy document.
type PolicyStatement struct {
 Effect string
 Action []string
 Principal map[string]string `json:",omitempty"`
 Resource *string `json:",omitempty"`
}



// PolicyWrapper encapsulates AWS Identity and Access Management (IAM) policy
 actions
// used in the examples.
// It contains an IAM service client that is used to perform policy actions.
type PolicyWrapper struct {
 IamClient *iam.Client
}



// ListPolicies gets up to maxPolicies policies.
func (wrapper PolicyWrapper) ListPolicies(maxPolicies int32) ([]types.Policy,
 error) {
 var policies []types.Policy
 result, err := wrapper.IamClient.ListPolicies(context.TODO(),
 &iam.ListPoliciesInput{
  MaxItems: aws.Int32(maxPolicies),
 })
 if err != nil {
  log.Printf("Couldn't list policies. Here's why: %v\n", err)
 } else {
  policies = result.Policies
 }
 return policies, err
}



// CreatePolicy creates a policy that grants a list of actions to the specified
 resource.
// PolicyDocument shows how to work with a policy document as a data structure
 and
```

```
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper PolicyWrapper) CreatePolicy(policyName string, actions []string,
  resourceArn string) (*types.Policy, error) {
 var policy *types.Policy
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(resourceArn),
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n",
 resourceArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreatePolicy(context.TODO(),
 &iam.CreatePolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
 })
 if err != nil {
  log.Printf("Couldn't create policy %v. Here's why: %v\n", policyName, err)
 } else {
  policy = result.Policy
 }
 return policy, err
}



// GetPolicy gets data about a policy.
func (wrapper PolicyWrapper) GetPolicy(policyArn string) (*types.Policy, error) {
 var policy *types.Policy
 result, err := wrapper.IamClient.GetPolicy(context.TODO(), &iam.GetPolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't get policy %v. Here's why: %v\n", policyArn, err)
 } else {
  policy = result.Policy
 }
```

```
 return policy, err
}



// DeletePolicy deletes a policy.
func (wrapper PolicyWrapper) DeletePolicy(policyArn string) error {
 _, err := wrapper.IamClient.DeletePolicy(context.TODO(), &iam.DeletePolicyInput{
  PolicyArn: aws.String(policyArn),
 })
 if err != nil {
  log.Printf("Couldn't delete policy %v. Here's why: %v\n", policyArn, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni del ruolo.

```
// RoleWrapper encapsulates AWS Identity and Access Management (IAM) role actions
// used in the examples.
// It contains an IAM service client that is used to perform role actions.
type RoleWrapper struct {
 IamClient *iam.Client
}



// ListRoles gets up to maxRoles roles.
func (wrapper RoleWrapper) ListRoles(maxRoles int32) ([]types.Role, error) {
 var roles []types.Role
 result, err := wrapper.IamClient.ListRoles(context.TODO(),
  &iam.ListRolesInput{MaxItems: aws.Int32(maxRoles)},
 )
 if err != nil {
  log.Printf("Couldn't list roles. Here's why: %v\n", err)
 } else {
  roles = result.Roles
 }
 return roles, err
}
```

```
// CreateRole creates a role that trusts a specified user. The trusted user can
 assume
// the role to acquire its permissions.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper RoleWrapper) CreateRole(roleName string, trustedUserArn string)
 (*types.Role, error) {
 var role *types.Role
 trustPolicy := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Principal: map[string]string{"AWS": trustedUserArn},
   Action: []string{"sts:AssumeRole"},
  }},
 }
 policyBytes, err := json.Marshal(trustPolicy)
 if err != nil {
  log.Printf("Couldn't create trust policy for %v. Here's why: %v\n",
 trustedUserArn, err)
  return nil, err
 }
 result, err := wrapper.IamClient.CreateRole(context.TODO(),
 &iam.CreateRoleInput{
  AssumeRolePolicyDocument: aws.String(string(policyBytes)),
  RoleName:                 aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't create role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// GetRole gets data about a role.
func (wrapper RoleWrapper) GetRole(roleName string) (*types.Role, error) {
 var role *types.Role
```

```
 result, err := wrapper.IamClient.GetRole(context.TODO(),
  &iam.GetRoleInput{RoleName: aws.String(roleName)})
 if err != nil {
  log.Printf("Couldn't get role %v. Here's why: %v\n", roleName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// CreateServiceLinkedRole creates a service-linked role that is owned by the
 specified service.
func (wrapper RoleWrapper) CreateServiceLinkedRole(serviceName string,
 description string) (*types.Role, error) {
 var role *types.Role
 result, err := wrapper.IamClient.CreateServiceLinkedRole(context.TODO(),
 &iam.CreateServiceLinkedRoleInput{
  AWSServiceName: aws.String(serviceName),
  Description:    aws.String(description),
 })
 if err != nil {
  log.Printf("Couldn't create service-linked role %v. Here's why: %v\n",
 serviceName, err)
 } else {
  role = result.Role
 }
 return role, err
}



// DeleteServiceLinkedRole deletes a service-linked role.
func (wrapper RoleWrapper) DeleteServiceLinkedRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteServiceLinkedRole(context.TODO(),
 &iam.DeleteServiceLinkedRoleInput{
  RoleName: aws.String(roleName)},
 )
 if err != nil {
  log.Printf("Couldn't delete service-linked role %v. Here's why: %v\n",
 roleName, err)
 }
 return err
```

```
}



// AttachRolePolicy attaches a policy to a role.
func (wrapper RoleWrapper) AttachRolePolicy(policyArn string, roleName string)
 error {
 _, err := wrapper.IamClient.AttachRolePolicy(context.TODO(),
 &iam.AttachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't attach policy %v to role %v. Here's why: %v\n", policyArn,
 roleName, err)
 }
 return err
}



// ListAttachedRolePolicies lists the policies that are attached to the specified
 role.
func (wrapper RoleWrapper) ListAttachedRolePolicies(roleName string)
 ([]types.AttachedPolicy, error) {
 var policies []types.AttachedPolicy
 result, err := wrapper.IamClient.ListAttachedRolePolicies(context.TODO(),
 &iam.ListAttachedRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list attached policies for role %v. Here's why: %v\n",
 roleName, err)
 } else {
  policies = result.AttachedPolicies
 }
 return policies, err
}



// DetachRolePolicy detaches a policy from a role.
func (wrapper RoleWrapper) DetachRolePolicy(roleName string, policyArn string)
 error {
```

```
 _, err := wrapper.IamClient.DetachRolePolicy(context.TODO(),
 &iam.DetachRolePolicyInput{
  PolicyArn: aws.String(policyArn),
  RoleName:  aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't detach policy from role %v. Here's why: %v\n", roleName,
 err)
 }
 return err
}



// ListRolePolicies lists the inline policies for a role.
func (wrapper RoleWrapper) ListRolePolicies(roleName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListRolePolicies(context.TODO(),
 &iam.ListRolePoliciesInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for role %v. Here's why: %v\n", roleName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteRole deletes a role. All attached policies must be detached before a
// role can be deleted.
func (wrapper RoleWrapper) DeleteRole(roleName string) error {
 _, err := wrapper.IamClient.DeleteRole(context.TODO(), &iam.DeleteRoleInput{
  RoleName: aws.String(roleName),
 })
 if err != nil {
  log.Printf("Couldn't delete role %v. Here's why: %v\n", roleName, err)
 }
 return err
}
```

Definisci una struttura che racchiude le azioni dell'utente.

```
// UserWrapper encapsulates user actions used in the examples.
// It contains an IAM service client that is used to perform user actions.
type UserWrapper struct {
 IamClient *iam.Client
}



// ListUsers gets up to maxUsers number of users.
func (wrapper UserWrapper) ListUsers(maxUsers int32) ([]types.User, error) {
 var users []types.User
 result, err := wrapper.IamClient.ListUsers(context.TODO(), &iam.ListUsersInput{
  MaxItems: aws.Int32(maxUsers),
 })
 if err != nil {
  log.Printf("Couldn't list users. Here's why: %v\n", err)
 } else {
  users = result.Users
 }
 return users, err
}



// GetUser gets data about a user.
func (wrapper UserWrapper) GetUser(userName string) (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.GetUser(context.TODO(), &iam.GetUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NoSuchEntityException:
    log.Printf("User %v does not exist.\n", userName)
    err = nil
```

```
   default:
    log.Printf("Couldn't get user %v. Here's why: %v\n", userName, err)
   }
  }
 } else {
  user = result.User
 }
 return user, err
}



// CreateUser creates a new user with the specified name.
func (wrapper UserWrapper) CreateUser(userName string) (*types.User, error) {
 var user *types.User
 result, err := wrapper.IamClient.CreateUser(context.TODO(),
 &iam.CreateUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
 } else {
  user = result.User
 }
 return user, err
}



// CreateUserPolicy adds an inline policy to a user. This example creates a
 policy that
// grants a list of actions on a specified role.
// PolicyDocument shows how to work with a policy document as a data structure
 and
// serialize it to JSON by using Go's JSON marshaler.
func (wrapper UserWrapper) CreateUserPolicy(userName string, policyName string,
 actions []string,
  roleArn string) error {
 policyDoc := PolicyDocument{
  Version:   "2012-10-17",
  Statement: []PolicyStatement{{
   Effect: "Allow",
   Action: actions,
   Resource: aws.String(roleArn),
```

```
  }},
 }
 policyBytes, err := json.Marshal(policyDoc)
 if err != nil {
  log.Printf("Couldn't create policy document for %v. Here's why: %v\n", roleArn,
 err)
  return err
 }
 _, err = wrapper.IamClient.PutUserPolicy(context.TODO(),
 &iam.PutUserPolicyInput{
  PolicyDocument: aws.String(string(policyBytes)),
  PolicyName:     aws.String(policyName),
  UserName:       aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't create policy for user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// ListUserPolicies lists the inline policies for the specified user.
func (wrapper UserWrapper) ListUserPolicies(userName string) ([]string, error) {
 var policies []string
 result, err := wrapper.IamClient.ListUserPolicies(context.TODO(),
 &iam.ListUserPoliciesInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list policies for user %v. Here's why: %v\n", userName,
 err)
 } else {
  policies = result.PolicyNames
 }
 return policies, err
}



// DeleteUserPolicy deletes an inline policy from a user.
func (wrapper UserWrapper) DeleteUserPolicy(userName string, policyName string)
 error {
```

```
 _, err := wrapper.IamClient.DeleteUserPolicy(context.TODO(),
 &iam.DeleteUserPolicyInput{
  PolicyName: aws.String(policyName),
  UserName:   aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete policy from user %v. Here's why: %v\n", userName,
 err)
 }
 return err
}



// DeleteUser deletes a user.
func (wrapper UserWrapper) DeleteUser(userName string) error {
 _, err := wrapper.IamClient.DeleteUser(context.TODO(), &iam.DeleteUserInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete user %v. Here's why: %v\n", userName, err)
 }
 return err
}



// CreateAccessKeyPair creates an access key for a user. The returned access key
 contains
// the ID and secret credentials needed to use the key.
func (wrapper UserWrapper) CreateAccessKeyPair(userName string)
 (*types.AccessKey, error) {
 var key *types.AccessKey
 result, err := wrapper.IamClient.CreateAccessKey(context.TODO(),
 &iam.CreateAccessKeyInput{
  UserName: aws.String(userName)})
 if err != nil {
  log.Printf("Couldn't create access key pair for user %v. Here's why: %v\n",
 userName, err)
 } else {
  key = result.AccessKey
 }
 return key, err
}
```

```
// DeleteAccessKey deletes an access key from a user.
func (wrapper UserWrapper) DeleteAccessKey(userName string, keyId string) error {
 _, err := wrapper.IamClient.DeleteAccessKey(context.TODO(),
 &iam.DeleteAccessKeyInput{
  AccessKeyId: aws.String(keyId),
  UserName:    aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't delete access key %v. Here's why: %v\n", keyId, err)
 }
 return err
}



// ListAccessKeys lists the access keys for the specified user.
func (wrapper UserWrapper) ListAccessKeys(userName string)
 ([]types.AccessKeyMetadata, error) {
 var keys []types.AccessKeyMetadata
 result, err := wrapper.IamClient.ListAccessKeys(context.TODO(),
 &iam.ListAccessKeysInput{
  UserName: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't list access keys for user %v. Here's why: %v\n", userName,
 err)
 } else {
  keys = result.AccessKeyMetadata
 }
 return keys, err
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy

- CreateRole
- CreateUser
- DeleteAccessChiave
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
/*
  To run this Java V2 code example, set up your development environment,
 including your credentials.

  For information, see this documentation topic:

  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html

  This example performs these operations:

  1. Creates a user that has no permissions.
  2. Creates a role and policy that grants Amazon S3 permissions.
  3. Creates a role.
  4. Grants the user permissions.
```

```
 5. Gets temporary credentials by assuming the role.  Creates an Amazon S3
Service client object with the temporary credentials.
 6. Deletes the resources.
*/

public class IAMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");
    public static final String PolicyDocument = "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}";

    public static String userArn;

    public static void main(String[] args) throws Exception {

        final String usage = """

                Usage:
                    <username> <policyName> <roleName> <roleSessionName>
<bucketName>\s

                Where:
                    username - The name of the IAM user to create.\s
                    policyName - The name of the policy to create.\s
                    roleName - The name of the role to create.\s
                    roleSessionName - The name of the session required for the
assumeRole operation.\s
                    bucketName - The name of the Amazon S3 bucket from which
objects are read.\s
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
```

```
        }

        String userName = args[0];
        String policyName = args[1];
        String roleName = args[2];
        String roleSessionName = args[3];
        String bucketName = args[4];

        Region region = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the AWS IAM example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 1. Create the IAM user.");
        User createUser = createIAMUser(iam, userName);

        System.out.println(DASHES);
        userArn = createUser.arn();

        AccessKey myKey = createIAMAccessKey(iam, userName);
        String accessKey = myKey.accessKeyId();
        String secretKey = myKey.secretAccessKey();
        String assumeRolePolicyDocument = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                " \"AWS\": \"" + userArn + "\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        System.out.println(assumeRolePolicyDocument);
        System.out.println(userName + " was successfully created.");
        System.out.println(DASHES);
        System.out.println("2. Creates a policy.");
        String polArn = createIAMPolicy(iam, policyName);
```

```
        System.out.println("The policy " + polArn + " was successfully
created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Creates a role.");
        TimeUnit.SECONDS.sleep(30);
        String roleArn = createIAMRole(iam, roleName, assumeRolePolicyDocument);
        System.out.println(roleArn + " was successfully created.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Grants the user permissions.");
        attachIAMRolePolicy(iam, roleName, polArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Wait for 30 secs so the resource is available");
        TimeUnit.SECONDS.sleep(30);
        System.out.println("5. Gets temporary credentials by assuming the
role.");
        System.out.println("Perform an Amazon S3 Service operation using the
temporary credentials.");
        assumeRole(roleArn, roleSessionName, bucketName, accessKey, secretKey);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6 Getting ready to delete the AWS resources");
        deleteKey(iam, userName, accessKey);
        deleteRole(iam, roleName, polArn);
        deleteIAMUser(iam, userName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("This IAM Scenario has successfully completed");
        System.out.println(DASHES);
    }

    public static AccessKey createIAMAccessKey(IamClient iam, String user) {
        try {
            CreateAccessKeyRequest request = CreateAccessKeyRequest.builder()
                    .userName(user)
                    .build();
```

```
            CreateAccessKeyResponse response = iam.createAccessKey(request);
            return response.accessKey();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static User createIAMUser(IamClient iam, String username) {
        try {
            // Create an IamWaiter object
            IamWaiter iamWaiter = iam.waiter();
            CreateUserRequest request = CreateUserRequest.builder()
                    .userName(username)
                    .build();

            // Wait until the user is created.
            CreateUserResponse response = iam.createUser(request);
            GetUserRequest userRequest = GetUserRequest.builder()
                    .userName(response.user().userName())
                    .build();

            WaiterResponse<GetUserResponse> waitUntilUserExists =
iamWaiter.waitUntilUserExists(userRequest);

waitUntilUserExists.matched().response().ifPresent(System.out::println);
            return response.user();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static String createIAMRole(IamClient iam, String rolename, String
json) {

        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(json)
```

```
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            System.out.println("The ARN of the role is " +
response.role().arn());
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static String createIAMPolicy(IamClient iam, String policyName) {
        try {
            // Create an IamWaiter object.
            IamWaiter iamWaiter = iam.waiter();
            CreatePolicyRequest request = CreatePolicyRequest.builder()
                    .policyName(policyName)
                    .policyDocument(PolicyDocument).build();

            CreatePolicyResponse response = iam.createPolicy(request);
            GetPolicyRequest polRequest = GetPolicyRequest.builder()
                    .policyArn(response.policy().arn())
                    .build();

            WaiterResponse<GetPolicyResponse> waitUntilPolicyExists =
iamWaiter.waitUntilPolicyExists(polRequest);

waitUntilPolicyExists.matched().response().ifPresent(System.out::println);
            return response.policy().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

    public static void attachIAMRolePolicy(IamClient iam, String roleName, String
policyArn) {
        try {
```

```
            ListAttachedRolePoliciesRequest request =
ListAttachedRolePoliciesRequest.builder()
                    .roleName(roleName)
                    .build();

            ListAttachedRolePoliciesResponse response =
iam.listAttachedRolePolicies(request);
            List<AttachedPolicy> attachedPolicies = response.attachedPolicies();
            String polArn;
            for (AttachedPolicy policy : attachedPolicies) {
                polArn = policy.policyArn();
                if (polArn.compareTo(policyArn) == 0) {
                    System.out.println(roleName + " policy is already attached to
this role.");
                    return;
                }
            }

            AttachRolePolicyRequest attachRequest =
AttachRolePolicyRequest.builder()
                    .roleName(roleName)
                    .policyArn(policyArn)
                    .build();

            iam.attachRolePolicy(attachRequest);
            System.out.println("Successfully attached policy " + policyArn + " to
role " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Invoke an Amazon S3 operation using the Assumed Role.
    public static void assumeRole(String roleArn, String roleSessionName, String
bucketName, String keyVal,
            String keySecret) {

        // Use the creds of the new IAM user that was created in this code
example.
        AwsBasicCredentials credentials = AwsBasicCredentials.create(keyVal,
keySecret);
        StsClient stsClient = StsClient.builder()
```

```
                .region(Region.US_EAST_1)

.credentialsProvider(StaticCredentialsProvider.create(credentials))
                .build();

        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();
            String key = myCreds.accessKeyId();
            String secKey = myCreds.secretAccessKey();
            String secToken = myCreds.sessionToken();

            // List all objects in an Amazon S3 bucket using the temp creds
retrieved by
            // invoking assumeRole.
            Region region = Region.US_EAST_1;
            S3Client s3 = S3Client.builder()
                    .credentialsProvider(

StaticCredentialsProvider.create(AwsSessionCredentials.create(key, secKey,
secToken)))
                    .region(region)
                    .build();

            System.out.println("Created a S3Client using temp credentials.");
            System.out.println("Listing objects in " + bucketName);
            ListObjectsRequest listObjects = ListObjectsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListObjectsResponse res = s3.listObjects(listObjects);
            List<S3Object> objects = res.contents();
            for (S3Object myValue : objects) {
                System.out.println("The name of the key is " + myValue.key());
                System.out.println("The owner is " + myValue.owner());
            }

        } catch (StsException e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }

    public static void deleteRole(IamClient iam, String roleName, String polArn)
{

        try {
            // First the policy needs to be detached.
            DetachRolePolicyRequest rolePolicyRequest =
DetachRolePolicyRequest.builder()
                    .policyArn(polArn)
                    .roleName(roleName)
                    .build();

            iam.detachRolePolicy(rolePolicyRequest);

            // Delete the policy.
            DeletePolicyRequest request = DeletePolicyRequest.builder()
                    .policyArn(polArn)
                    .build();

            iam.deletePolicy(request);
            System.out.println("*** Successfully deleted " + polArn);

            // Delete the role.
            DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                    .roleName(roleName)
                    .build();

            iam.deleteRole(roleRequest);
            System.out.println("*** Successfully deleted " + roleName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteKey(IamClient iam, String username, String
accessKey) {
        try {
            DeleteAccessKeyRequest request = DeleteAccessKeyRequest.builder()
                    .accessKeyId(accessKey)
```

```
                    .userName(username)
                    .build();

            iam.deleteAccessKey(request);
            System.out.println("Successfully deleted access key " + accessKey +
                    " from user " + username);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteIAMUser(IamClient iam, String userName) {
        try {
            DeleteUserRequest request = DeleteUserRequest.builder()
                    .userName(userName)
                    .build();

            iam.deleteUser(request);
            System.out.println("*** Successfully deleted " + userName);

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole

- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import {
  CreateUserCommand,
  GetUserCommand,
  CreateAccessKeyCommand,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
  DeleteAccessKeyCommand,
  DeleteUserCommand,
  DeleteRoleCommand,
  DeletePolicyCommand,
  DetachRolePolicyCommand,
  IAMClient,
} from "@aws-sdk/client-iam";
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
import { AssumeRoleCommand, STSClient } from "@aws-sdk/client-sts";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { ScenarioInput } from "@aws-doc-sdk-examples/lib/scenario/index.js";

// Set the parameters.
const iamClient = new IAMClient({});
```

```
const userName = "test_name";
const policyName = "test_policy";
const roleName = "test_role";

/**
 * Create a new IAM user. If the user already exists, give
 * the option to delete and re-create it.
 * @param {string} name
 */
export const createUser = async (name, confirmAll = false) => {
  try {
    const { User } = await iamClient.send(
      new GetUserCommand({ UserName: name }),
    );
    const input = new ScenarioInput(
      "deleteUser",
      "Do you want to delete and remake this user?",
      { type: "confirm" },
    );
    const deleteUser = await input.handle({}, { confirmAll });
    // If the user exists, and you want to delete it, delete the user
    // and then create it again.
    if (deleteUser) {
      await iamClient.send(new DeleteUserCommand({ UserName: User.UserName }));
      await iamClient.send(new CreateUserCommand({ UserName: name }));
    } else {
      console.warn(
        `${name} already exists. The scenario may not work as expected.`,
      );
      return User;
    }
  } catch (caught) {
    // If there is no user by that name, create one.
    if (caught instanceof Error && caught.name === "NoSuchEntityException") {
      const { User } = await iamClient.send(
        new CreateUserCommand({ UserName: name }),
      );
      return User;
    } else {
      throw caught;
    }
  }
};
```

```
export const main = async (confirmAll = false) => {
  // Create a user. The user has no permissions by default.
  const User = await createUser(userName, confirmAll);

  if (!User) {
    throw new Error("User not created");
  }

  // Create an access key. This key is used to authenticate the new user to
  // Amazon Simple Storage Service (Amazon S3) and AWS Security Token Service
 (AWS STS).
  // It's not best practice to use access keys. For more information, see
 https://aws.amazon.com/iam/resources/best-practices/.
  const createAccessKeyResponse = await iamClient.send(
    new CreateAccessKeyCommand({ UserName: userName }),
  );

  if (
    !createAccessKeyResponse.AccessKey?.AccessKeyId ||
    !createAccessKeyResponse.AccessKey?.SecretAccessKey
  ) {
    throw new Error("Access key not created");
  }

  const {
    AccessKey: { AccessKeyId, SecretAccessKey },
  } = createAccessKeyResponse;

  let s3Client = new S3Client({
    credentials: {
      accessKeyId: AccessKeyId,
      secretAccessKey: SecretAccessKey,
    },
  });

  // Retry the list buckets operation until it succeeds. InvalidAccessKeyId is
  // thrown while the user and access keys are still stabilizing.
  await retry({ intervalInMs: 1000, maxRetries: 300 }, async () => {
    try {
      return await listBuckets(s3Client);
    } catch (err) {
      if (err instanceof Error && err.name === "InvalidAccessKeyId") {
        throw err;
      }
```

```
    }
  });

  // Retry the create role operation until it succeeds. A MalformedPolicyDocument
error
  // is thrown while the user and access keys are still stabilizing.
  const { Role } = await retry(
    {
      intervalInMs: 2000,
      maxRetries: 60,
    },
    () =>
      iamClient.send(
        new CreateRoleCommand({
          AssumeRolePolicyDocument: JSON.stringify({
            Version: "2012-10-17",
            Statement: [
              {
                Effect: "Allow",
                Principal: {
                  // Allow the previously created user to assume this role.
                  AWS: User.Arn,
                },
                Action: "sts:AssumeRole",
              },
            ],
          }),
          RoleName: roleName,
        }),
      ),
  );

  if (!Role) {
    throw new Error("Role not created");
  }

  // Create a policy that allows the user to list S3 buckets.
  const { Policy: listBucketPolicy } = await iamClient.send(
    new CreatePolicyCommand({
      PolicyDocument: JSON.stringify({
        Version: "2012-10-17",
        Statement: [
          {
            Effect: "Allow",
```

```
          Action: ["s3:ListAllMyBuckets"],
          Resource: "*",
        },
      ],
    }),
    PolicyName: policyName,
  }),
);

if (!listBucketPolicy) {
  throw new Error("Policy not created");
}

// Attach the policy granting the 's3:ListAllMyBuckets' action to the role.
await iamClient.send(
  new AttachRolePolicyCommand({
    PolicyArn: listBucketPolicy.Arn,
    RoleName: Role.RoleName,
  }),
);

// Assume the role.
const stsClient = new STSClient({
  credentials: {
    accessKeyId: AccessKeyId,
    secretAccessKey: SecretAccessKey,
  },
});

// Retry the assume role operation until it succeeds.
const { Credentials } = await retry(
  { intervalInMs: 2000, maxRetries: 60 },
  () =>
    stsClient.send(
      new AssumeRoleCommand({
        RoleArn: Role.Arn,
        RoleSessionName: `iamBasicScenarioSession-${Math.floor(
          Math.random() * 1000000,
        )}`,
        DurationSeconds: 900,
      }),
    ),
);
```

```
  if (!Credentials?.AccessKeyId || !Credentials?.SecretAccessKey) {
    throw new Error("Credentials not created");
  }

  s3Client = new S3Client({
    credentials: {
      accessKeyId: Credentials.AccessKeyId,
      secretAccessKey: Credentials.SecretAccessKey,
      sessionToken: Credentials.SessionToken,
    },
  });

  // List the S3 buckets again.
  // Retry the list buckets operation until it succeeds. AccessDenied might
  // be thrown while the role policy is still stabilizing.
  await retry({ intervalInMs: 2000, maxRetries: 60 }, () =>
    listBuckets(s3Client),
  );

  // Clean up.
  await iamClient.send(
    new DetachRolePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeletePolicyCommand({
      PolicyArn: listBucketPolicy.Arn,
    }),
  );

  await iamClient.send(
    new DeleteRoleCommand({
      RoleName: Role.RoleName,
    }),
  );

  await iamClient.send(
    new DeleteAccessKeyCommand({
      UserName: userName,
      AccessKeyId,
    }),
```

```
  );

  await iamClient.send(
    new DeleteUserCommand({
      UserName: userName,
    }),
  );
};

/**
 *
 * @param {S3Client} s3Client
 */
const listBuckets = async (s3Client) => {
  const { Buckets } = await s3Client.send(new ListBucketsCommand({}));

  if (!Buckets) {
    throw new Error("Buckets not listed");
  }

  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica

- PutUserPolitica

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <username> <policyName> <roleName> <roleSessionName> <fileLocation>
 <bucketName>

    Where:
        username - The name of the IAM user to create.
        policyName - The name of the policy to create.
        roleName - The name of the role to create.
        roleSessionName - The name of the session required for the assumeRole
 operation.
        fileLocation - The file location to the JSON required to create the role
 (see Readme).
        bucketName - The name of the Amazon S3 bucket from which objects are
 read.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val userName = args[0]
    val policyName = args[1]
    val roleName = args[2]
    val roleSessionName = args[3]
    val fileLocation = args[4]
```

```
    val bucketName = args[5]

    createUser(userName)
    println("$userName was successfully created.")

    val polArn = createPolicy(policyName)
    println("The policy $polArn was successfully created.")

    val roleArn = createRole(roleName, fileLocation)
    println("$roleArn was successfully created.")
    attachRolePolicy(roleName, polArn)

    println("*** Wait for 1 MIN so the resource is available.")
    delay(60000)
    assumeGivenRole(roleArn, roleSessionName, bucketName)

    println("*** Getting ready to delete the AWS resources.")
    deleteRole(roleName, polArn)
    deleteUser(userName)
    println("This IAM Scenario has successfully completed.")
}

suspend fun createUser(usernameVal: String?): String? {
    val request =
        CreateUserRequest {
            userName = usernameVal
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createUser(request)
        return response.user?.userName
    }
}

suspend fun createPolicy(policyNameVal: String?): String {
    val policyDocumentValue: String =
        "{" +
            "  \"Version\": \"2012-10-17\"," +
            "  \"Statement\": [" +
            "    {" +
            "        \"Effect\": \"Allow\"," +
            "        \"Action\": [" +
            "            \"s3:*\"" +
            "       ]," +
```

```
            "       \"Resource\": \"*\"" +
            "    }" +
            "   ]" +
            "}"

    val request =
        CreatePolicyRequest {
            policyName = policyNameVal
            policyDocument = policyDocumentValue
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createPolicy(request)
        return response.policy?.arn.toString()
    }
}

suspend fun createRole(
    rolenameVal: String?,
    fileLocation: String?
): String? {
    val jsonObject = fileLocation?.let { readJsonSimpleDemo(it) } as JSONObject

    val request =
        CreateRoleRequest {
            roleName = rolenameVal
            assumeRolePolicyDocument = jsonObject.toJSONString()
            description = "Created using the AWS SDK for Kotlin"
        }

    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.createRole(request)
        return response.role?.arn
    }
}

suspend fun attachRolePolicy(
    roleNameVal: String,
    policyArnVal: String
) {
    val request =
        ListAttachedRolePoliciesRequest {
            roleName = roleNameVal
        }
```

```
    IamClient { region = "AWS_GLOBAL" }.use { iamClient ->
        val response = iamClient.listAttachedRolePolicies(request)
        val attachedPolicies = response.attachedPolicies

        // Ensure that the policy is not attached to this role.
        val checkStatus: Int
        if (attachedPolicies != null) {
            checkStatus = checkMyList(attachedPolicies, policyArnVal)
            if (checkStatus == -1) {
                return
            }
        }

        val policyRequest =
            AttachRolePolicyRequest {
                roleName = roleNameVal
                policyArn = policyArnVal
            }
        iamClient.attachRolePolicy(policyRequest)
        println("Successfully attached policy $policyArnVal to role
 $roleNameVal")
    }
}

fun checkMyList(
    attachedPolicies: List<AttachedPolicy>,
    policyArnVal: String
): Int {
    for (policy in attachedPolicies) {
        val polArn = policy.policyArn.toString()

        if (polArn.compareTo(policyArnVal) == 0) {
            println("The policy is already attached to this role.")
            return -1
        }
    }
    return 0
}

suspend fun assumeGivenRole(
    roleArnVal: String?,
    roleSessionNameVal: String?,
    bucketName: String
```

```
) {
    val stsClient =
        StsClient {
            region = "us-east-1"
        }

    val roleRequest =
        AssumeRoleRequest {
            roleArn = roleArnVal
            roleSessionName = roleSessionNameVal
        }

    val roleResponse = stsClient.assumeRole(roleRequest)
    val myCreds = roleResponse.credentials
    val key = myCreds?.accessKeyId
    val secKey = myCreds?.secretAccessKey
    val secToken = myCreds?.sessionToken

    val staticCredentials =
        StaticCredentialsProvider {
            accessKeyId = key
            secretAccessKey = secKey
            sessionToken = secToken
        }

    // List all objects in an Amazon S3 bucket using the temp creds.
    val s3 =
        S3Client {
            credentialsProvider = staticCredentials
            region = "us-east-1"
        }

    println("Created a S3Client using temp credentials.")
    println("Listing objects in $bucketName")

    val listObjects =
        ListObjectsRequest {
            bucket = bucketName
        }

    val response = s3.listObjects(listObjects)
    response.contents?.forEach { myObject ->
        println("The name of the key is ${myObject.key}")
        println("The owner is ${myObject.owner}")
```

```
    }
}

suspend fun deleteRole(
    roleNameVal: String,
    polArn: String
) {
    val iam = IamClient { region = "AWS_GLOBAL" }

    // First the policy needs to be detached.
    val rolePolicyRequest =
        DetachRolePolicyRequest {
            policyArn = polArn
            roleName = roleNameVal
        }

    iam.detachRolePolicy(rolePolicyRequest)

    // Delete the policy.
    val request =
        DeletePolicyRequest {
            policyArn = polArn
        }

    iam.deletePolicy(request)
    println("*** Successfully deleted $polArn")

    // Delete the role.
    val roleRequest =
        DeleteRoleRequest {
            roleName = roleNameVal
        }

    iam.deleteRole(roleRequest)
    println("*** Successfully deleted $roleNameVal")
}

suspend fun deleteUser(userNameVal: String) {
    val iam = IamClient { region = "AWS_GLOBAL" }
    val request =
        DeleteUserRequest {
            userName = userNameVal
        }
```

```
    iam.deleteUser(request)
    println("*** Successfully deleted $userNameVal")
}

@Throws(java.lang.Exception::class)
fun readJsonSimpleDemo(filename: String): Any? {
    val reader = FileReader(filename)
    val jsonParser = JSONParser()
    return jsonParser.parse(reader)
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

PHP

SDK per PHP

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace Iam\Basics;

require 'vendor/autoload.php';

use Aws\Credentials\Credentials;
use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;
use Iam\IAMService;

echo("\n");
echo("---------------------------------------\n");
print("Welcome to the IAM getting started demo using PHP!\n");
echo("---------------------------------------\n");

$uuid = uniqid();
$service = new IAMService();

$user = $service->createUser("iam_demo_user_$uuid");
echo "Created user with the arn: {$user['Arn']}\n";

$key = $service->createAccessKey($user['UserName']);
$assumeRolePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{$user['Arn']}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }";
$assumeRoleRole = $service->createRole("iam_demo_role_$uuid",
 $assumeRolePolicyDocument);
echo "Created role: {$assumeRoleRole['RoleName']}\n";

$listAllBucketsPolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
}";
$listAllBucketsPolicy = $service->createPolicy("iam_demo_policy_$uuid",
 $listAllBucketsPolicyDocument);
```

```
echo "Created policy: {$listAllBucketsPolicy['PolicyName']}\n";

$service->attachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);

$inlinePolicyDocument = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{$assumeRoleRole['Arn']}\"}]
}";
$inlinePolicy = $service->createUserPolicy("iam_demo_inline_policy_$uuid",
 $inlinePolicyDocument, $user['UserName']);
//First, fail to list the buckets with the user
$credentials = new Credentials($key['AccessKeyId'], $key['SecretAccessKey']);
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
try {
    $s3Client->listBuckets([
    ]);
    echo "this should not run";
} catch (S3Exception $exception) {
    echo "successfully failed!\n";
}

$stsClient = new StsClient(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $credentials]);
sleep(10);
$assumedRole = $stsClient->assumeRole([
    'RoleArn' => $assumeRoleRole['Arn'],
    'RoleSessionName' => "DemoAssumeRoleSession_$uuid",
]);
$assumedCredentials = [
    'key' => $assumedRole['Credentials']['AccessKeyId'],
    'secret' => $assumedRole['Credentials']['SecretAccessKey'],
    'token' => $assumedRole['Credentials']['SessionToken'],
];
$s3Client = new S3Client(['region' => 'us-west-2', 'version' => 'latest',
 'credentials' => $assumedCredentials]);
try {
    $s3Client->listBuckets([]);
    echo "this should now run!\n";
} catch (S3Exception $exception) {
```

```
    echo "this should now not fail\n";
}

$service->detachRolePolicy($assumeRoleRole['RoleName'],
 $listAllBucketsPolicy['Arn']);
$deletePolicy = $service->deletePolicy($listAllBucketsPolicy['Arn']);
echo "Delete policy: {$listAllBucketsPolicy['PolicyName']}\n";
$deletedRole = $service->deleteRole($assumeRoleRole['Arn']);
echo "Deleted role: {$assumeRoleRole['RoleName']}\n";
$deletedKey = $service->deleteAccessKey($key['AccessKeyId'], $user['UserName']);
$deletedUser = $service->deleteUser($user['UserName']);
echo "Delete user: {$user['UserName']}\n";
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for PHP .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
import json
import sys
import time
from uuid import uuid4

import boto3
from botocore.exceptions import ClientError


def progress_bar(seconds):
    """Shows a simple progress bar in the command window."""
    for _ in range(seconds):
        time.sleep(1)
        print(".", end="")
        sys.stdout.flush()
    print()


def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
    Creates an inline policy for the user that lets the user assume the role.
```

```
    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
resource
                         that has permissions to create users, roles, and
policies
                         in the account.
    :return: The newly created user, user key, and role.
    """
    try:
        user = iam_resource.create_user(UserName=f"demo-user-{uuid4()}")
        print(f"Created user {user.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a user for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user_key = user.create_access_key_pair()
        print(f"Created access key pair for user.")
    except ClientError as error:
        print(
            f"Couldn't create access keys for user {user.name}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    try:
        role = iam_resource.create_role(
            RoleName=f"demo-role-{uuid4()}",
            AssumeRolePolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {"AWS": user.arn},
                            "Action": "sts:AssumeRole",
                        }
                    ],
                }
```

```
            ),
        )
        print(f"Created role {role.name}.")
    except ClientError as error:
        print(
            f"Couldn't create a role for the demo. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        policy = iam_resource.create_policy(
            PolicyName=f"demo-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "s3:ListAllMyBuckets",
                            "Resource": "arn:aws:s3:::*",
                        }
                    ],
                }
            ),
        )
        role.attach_policy(PolicyArn=policy.arn)
        print(f"Created policy {policy.policy_name} and attached it to the
role.")
    except ClientError as error:
        print(
            f"Couldn't create a policy and attach it to role {role.name}. Here's
why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        user.create_policy(
            PolicyName=f"demo-user-policy-{uuid4()}",
            PolicyDocument=json.dumps(
                {
                    "Version": "2012-10-17",
                    "Statement": [
```

```
                        {
                            "Effect": "Allow",
                            "Action": "sts:AssumeRole",
                            "Resource": role.arn,
                        }
                    ],
                }
            ),
        )
        print(
            f"Created an inline policy for {user.name} that lets the user assume
 "
            f"the role."
        )
    except ClientError as error:
        print(
            f"Couldn't create an inline policy for user {user.name}. Here's why:
 "
            f"{error.response['Error']['Message']}"
        )
        raise

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, role


def show_access_denied_without_role(user_key):
    """
    Shows that listing buckets without first assuming the role is not allowed.

    :param user_key: The key of the user created during setup. This user does not
                     have permission to list buckets in the account.
    """
    print(f"Try to list buckets without first assuming the role.")
    s3_denied_resource = boto3.resource(
        "s3", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        for bucket in s3_denied_resource.buckets.all():
            print(bucket.name)
```

```
        raise RuntimeError("Expected to get AccessDenied error when listing
 buckets!")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Attempt to list buckets with no permissions: AccessDenied.")
        else:
            raise


def list_buckets_from_assumed_role(user_key, assume_role_arn, session_name):
    """
    Assumes a role that grants permission to list the Amazon S3 buckets in the
 account.
    Uses the temporary credentials from the role to list the buckets that are
 owned
    by the assumed role's account.

    :param user_key: The access key of a user that has permission to assume the
 role.
    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    """
    sts_client = boto3.client(
        "sts", aws_access_key_id=user_key.id,
 aws_secret_access_key=user_key.secret
    )
    try:
        response = sts_client.assume_role(
            RoleArn=assume_role_arn, RoleSessionName=session_name
        )
        temp_credentials = response["Credentials"]
        print(f"Assumed role {assume_role_arn} and got temporary credentials.")
    except ClientError as error:
        print(
            f"Couldn't assume role {assume_role_arn}. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    # Create an S3 resource that can access the account with the temporary
 credentials.
    s3_resource = boto3.resource(
        "s3",
```

```
            aws_access_key_id=temp_credentials["AccessKeyId"],
            aws_secret_access_key=temp_credentials["SecretAccessKey"],
            aws_session_token=temp_credentials["SessionToken"],
        )
        print(f"Listing buckets for the assumed role's account:")
        try:
            for bucket in s3_resource.buckets.all():
                print(bucket.name)
        except ClientError as error:
            print(
                f"Couldn't list buckets for the account. Here's why: "
                f"{error.response['Error']['Message']}"
            )
            raise




def teardown(user, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    try:
        for attached in role.attached_policies.all():
            policy_name = attached.policy_name
            role.detach_policy(PolicyArn=attached.arn)
            attached.delete()
            print(f"Detached and deleted {policy_name}.")
        role.delete()
        print(f"Deleted {role.name}.")
    except ClientError as error:
        print(
            "Couldn't detach policy, delete policy, or delete role. Here's why: "
            f"{error.response['Error']['Message']}"
        )
        raise

    try:
        for user_pol in user.policies.all():
            user_pol.delete()
            print("Deleted inline user policy.")
```

```
        for key in user.access_keys.all():
            key.delete()
            print("Deleted user's access key.")
        user.delete()
        print(f"Deleted {user.name}.")
    except ClientError as error:
        print(
            "Couldn't delete user policy or delete user. Here's why: "
            f"{error.response['Error']['Message']}"
        )


def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(f"Welcome to the IAM create user and assume role demo.")
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user = None
    role = None
    try:
        user, user_key, role = setup(iam_resource)
        print(f"Created {user.name} and {role.name}.")
        show_access_denied_without_role(user_key)
        list_buckets_from_assumed_role(user_key, role.arn,
 "AssumeRoleDemoSession")
    except Exception:
        print("Something went wrong!")
    finally:
        if user is not None and role is not None:
            teardown(user, role)
        print("Thanks for watching!")


if __name__ == "__main__":
    usage_demo()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AttachRolePolitica
  - CreateAccessChiave

- CreatePolicy
- CreateRole
- CreateUser
- DeleteAccessChiave
- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolitica
- DetachRolePolitica
- PutUserPolitica

Ruby

SDK per Ruby

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un utente IAM che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo. Dopo aver assunto il ruolo, utilizza le credenziali temporanee per elencare i bucket per l'account.

```
# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_client

  # @param [Aws::IAM::Client] iam_client: The AWS IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Waits for the specified number of seconds.
  #
```

```
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts("Give AWS time to propagate resources...")
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_client.create_user(user_name: user_name).user
    @logger.info("Created demo user named #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Tried and failed to create demo user.")
    @logger.info("\t#{e.code}: #{e.message}")
    @logger.info("\nCan't continue the demo without a user!")
    raise
  else
    user
  end

  # Creates an access key for a user.
  #
  # @param user [Aws::IAM::User] The user that owns the key.
  # @return [Aws::IAM::AccessKeyPair] The newly created access key.
  def create_access_key_pair(user)
    user_key = @iam_client.create_access_key(user_name:
user.user_name).access_key
    @logger.info("Created accesskey pair for user #{user.user_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create access keys for user #{user.user_name}.")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    user_key
  end

  # Creates a role that can be assumed by a user.
  #
  # @param role_name [String] The name to give the role.
  # @param user [Aws::IAM::User] The user who is granted permission to assume the
role.
  # @return [Aws::IAM::Role] The newly created role.
```

```
  def create_role(role_name, user)
    trust_policy = {
      Version: "2012-10-17",
      Statement: [{
                   Effect: "Allow",
                   Principal: {'AWS': user.arn},
                   Action: "sts:AssumeRole"
                 }]
    }.to_json
    role = @iam_client.create_role(
      role_name: role_name,
      assume_role_policy_document: trust_policy
    ).role
    @logger.info("Created role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a role for the demo. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  else
    role
  end

  # Creates a policy that grants permission to list S3 buckets in the account,
and
  # then attaches the policy to a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param role [Aws::IAM::Role] The role that the policy is attached to.
  # @return [Aws::IAM::Policy] The newly created policy.
  def create_and_attach_role_policy(policy_name, role)
    policy_document = {
      Version: "2012-10-17",
      Statement: [{
                   Effect: "Allow",
                   Action: "s3:ListAllMyBuckets",
                   Resource: "arn:aws:s3:::*"
                 }]
    }.to_json
    policy = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document
    ).policy
    @iam_client.attach_role_policy(
      role_name: role.role_name,
```

```
      policy_arn: policy.arn
    )
    @logger.info("Created policy #{policy.policy_name} and attached it to role
#{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a policy and attach it to role
#{role.role_name}. Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an inline policy for a user that lets the user assume a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param user [Aws::IAM::User] The user that owns the policy.
  # @param role [Aws::IAM::Role] The role that can be assumed.
  # @return [Aws::IAM::UserPolicy] The newly created policy.
  def create_user_policy(policy_name, user, role)
    policy_document = {
      Version: "2012-10-17",
      Statement: [{
                   Effect: "Allow",
                   Action: "sts:AssumeRole",
                   Resource: role.arn
                 }]
    }.to_json
    @iam_client.put_user_policy(
      user_name: user.user_name,
      policy_name: policy_name,
      policy_document: policy_document
    )
    puts("Created an inline policy for #{user.user_name} that lets the user
assume role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create an inline policy for user #{user.user_name}.
Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an Amazon S3 resource with specified credentials. This is separated
into a
  # factory function so that it can be mocked for unit testing.
  #
```

```
 # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
 def create_s3_resource(credentials)
   Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
 end

 # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
 # Because the resource uses credentials with limited access, it may not be able
to
 # list the S3 buckets.
 #
 # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
 def list_buckets(s3_resource)
   count = 10
   s3_resource.buckets.each do |bucket|
     @logger.info "\t#{bucket.name}"
     count -= 1
     break if count.zero?
   end
 rescue Aws::Errors::ServiceError => e
   if e.code == "AccessDenied"
     puts("Attempt to list buckets with no permissions: AccessDenied.")
   else
     @logger.info("Couldn't list buckets for the account. Here's why: ")
     @logger.info("\t#{e.code}: #{e.message}")
     raise
   end
 end

 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
```

```
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: "create-use-assume-role-scenario"
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end

 # Deletes a role. If the role has policies attached, they are detached and
 # deleted before the role is deleted.
 #
 # @param role_name [String] The name of the role to delete.
 def delete_role(role_name)
   @iam_client.list_attached_role_policies(role_name:
role_name).attached_policies.each do |policy|
     @iam_client.detach_role_policy(role_name: role_name, policy_arn:
policy.policy_arn)
     @iam_client.delete_policy(policy_arn: policy.policy_arn)
     @logger.info("Detached and deleted policy #{policy.policy_name}.")
   end
   @iam_client.delete_role({ role_name: role_name })
   @logger.info("Role deleted: #{role_name}.")
 rescue Aws::Errors::ServiceError => e
   @logger.info("Couldn't detach policies and delete role #{role.name}. Here's
why:")
   @logger.info("\t#{e.code}: #{e.message}")
   raise
 end

 # Deletes a user. If the user has inline policies or access keys, they are
deleted
 # before the user is deleted.
 #
 # @param user [Aws::IAM::User] The user to delete.
 def delete_user(user_name)
   user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
```

```
    user.each do |key|
      @iam_client.delete_access_key({ access_key_id: key.access_key_id,
 user_name: user_name })
      @logger.info("Deleted access key #{key.access_key_id} for user
 '#{user_name}'.")
    end

    @iam_client.delete_user(user_name: user_name)
    @logger.info("Deleted user '#{user_name}'.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting user '#{user_name}': #{e.message}")
  end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the IAM create a user and assume a role demo!")
  puts("-" * 88)
  user = scenario.create_user("doc-example-user-#{Random.uuid}")
  user_key = scenario.create_access_key_pair(user)
  scenario.wait(10)
  role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
  scenario.create_and_attach_role_policy("doc-example-role-policy-
#{Random.uuid}", role)
  scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user,
 role)
  scenario.wait(10)
  puts("Try to list buckets with credentials for a user who has no permissions.")
  puts("Expect AccessDenied from this call.")
  scenario.list_buckets(
    scenario.create_s3_resource(Aws::Credentials.new(user_key.access_key_id,
 user_key.secret_access_key)))
  puts("Now, assume the role that grants permission.")
  temp_credentials = scenario.assume_role(
    role.arn, scenario.create_sts_client(user_key.access_key_id,
 user_key.secret_access_key))
  puts("Here are your buckets:")
  scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
  puts("Deleting role '#{role.role_name}' and attached policies.")
  scenario.delete_role(role.role_name)
  puts("Deleting user '#{user.user_name}', policies, and keys.")
  scenario.delete_user(user.user_name)
  puts("Thanks for watching!")
```

```
  puts("-" * 88)
rescue Aws::Errors::ServiceError => e
  puts("Something went wrong with the demo.")
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Client.new)) if
 $PROGRAM_NAME == __FILE__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Ruby .
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_iam::Error as iamError;
use aws_sdk_iam::{config::Credentials as iamCredentials, config::Region, Client
 as iamClient};
use aws_sdk_s3::Client as s3Client;
use aws_sdk_sts::Client as stsClient;
use tokio::time::{sleep, Duration};
use uuid::Uuid;

#[tokio::main]
async fn main() -> Result<(), iamError> {
    let (client, uuid, list_all_buckets_policy_document, inline_policy_document)
 =
        initialize_variables().await;

    if let Err(e) = run_iam_operations(
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
    .await
    {
        println!("{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (iamClient, String, String, String) {
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));

    let shared_config =
 aws_config::from_env().region(region_provider).load().await;
    let client = iamClient::new(&shared_config);
    let uuid = Uuid::new_v4().to_string();

    let list_all_buckets_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"s3:ListAllMyBuckets\",
                    \"Resource\": \"arn:aws:s3:::*\"}]
```

```
    }"
    .to_string();
    let inline_policy_document = "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Action\": \"sts:AssumeRole\",
                    \"Resource\": \"{}\"}]
    }"
    .to_string();

    (
        client,
        uuid,
        list_all_buckets_policy_document,
        inline_policy_document,
    )
}

async fn run_iam_operations(
    client: iamClient,
    uuid: String,
    list_all_buckets_policy_document: String,
    inline_policy_document: String,
) -> Result<(), iamError> {
    let user = iam_service::create_user(&client, &format!("{}{}",
 "iam_demo_user_", uuid)).await?;
    println!("Created the user with the name: {}", user.user_name());
    let key = iam_service::create_access_key(&client, user.user_name()).await?;

    let assume_role_policy_document = "{
        \"Version\": \"2012-10-17\",
                \"Statement\": [{
                    \"Effect\": \"Allow\",
                    \"Principal\": {\"AWS\": \"{}\"},
                    \"Action\": \"sts:AssumeRole\"
                }]
            }"
    .to_string()
    .replace("{}", user.arn());

    let assume_role_role = iam_service::create_role(
        &client,
        &format!("{}{}", "iam_demo_role_", uuid),
```

```
        &assume_role_policy_document,
    )
    .await?;
    println!("Created the role with the ARN: {}", assume_role_role.arn());

    let list_all_buckets_policy = iam_service::create_policy(
        &client,
        &format!("{}{}", "iam_demo_policy_", uuid),
        &list_all_buckets_policy_document,
    )
    .await?;
    println!(
        "Created policy: {}",
        list_all_buckets_policy.policy_name.as_ref().unwrap()
    );

    let attach_role_policy_result =
        iam_service::attach_role_policy(&client, &assume_role_role,
&list_all_buckets_policy)
            .await?;
    println!(
        "Attached the policy to the role: {:?}",
        attach_role_policy_result
    );

    let inline_policy_name = format!("{}{}", "iam_demo_inline_policy_", uuid);
    let inline_policy_document = inline_policy_document.replace("{}",
assume_role_role.arn());
    iam_service::create_user_policy(&client, &user, &inline_policy_name,
&inline_policy_document)
        .await?;
    println!("Created inline policy.");

    //First, fail to list the buckets with the user.
    let creds = iamCredentials::from_keys(key.access_key_id(),
key.secret_access_key(), None);
    let fail_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    println!("Fail config: {:?}", fail_config);
    let fail_client: s3Client = s3Client::new(&fail_config);
    match fail_client.list_buckets().send().await {
        Ok(e) => {
```

```
            println!("This should not run. {:?}", e);
        }
        Err(e) => {
            println!("Successfully failed with error: {:?}", e)
        }
    }

    let sts_config = aws_config::from_env()
        .credentials_provider(creds.clone())
        .load()
        .await;
    let sts_client: stsClient = stsClient::new(&sts_config);
    sleep(Duration::from_secs(10)).await;
    let assumed_role = sts_client
        .assume_role()
        .role_arn(assume_role_role.arn())
        .role_session_name(&format!("{}{}", "iam_demo_assumerole_session_",
uuid))
        .send()
        .await;
    println!("Assumed role: {:?}", assumed_role);
    sleep(Duration::from_secs(10)).await;

    let assumed_credentials = iamCredentials::from_keys(
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .access_key_id(),
        assumed_role
            .as_ref()
            .unwrap()
            .credentials
            .as_ref()
            .unwrap()
            .secret_access_key(),
        Some(
            assumed_role
                .as_ref()
                .unwrap()
                .credentials
                .as_ref()
```

```
                .unwrap()
                .session_token
                .clone(),
        ),
    );

    let succeed_config = aws_config::from_env()
        .credentials_provider(assumed_credentials)
        .load()
        .await;
    println!("succeed config: {:?}", succeed_config);
    let succeed_client: s3Client = s3Client::new(&succeed_config);
    sleep(Duration::from_secs(10)).await;
    match succeed_client.list_buckets().send().await {
        Ok(_) => {
            println!("This should now run successfully.")
        }
        Err(e) => {
            println!("This should not run. {:?}", e);
            panic!()
        }
    }

    //Clean up.
    iam_service::detach_role_policy(
        &client,
        assume_role_role.role_name(),
        list_all_buckets_policy.arn().unwrap_or_default(),
    )
    .await?;
    iam_service::delete_policy(&client, list_all_buckets_policy).await?;
    iam_service::delete_role(&client, &assume_role_role).await?;
    println!("Deleted role {}", assume_role_role.role_name());
    iam_service::delete_access_key(&client, &user, &key).await?;
    println!("Deleted key for {}", key.user_name());
    iam_service::delete_user_policy(&client, &user, &inline_policy_name).await?;
    println!("Deleted inline user policy: {}", inline_policy_name);
    iam_service::delete_user(&client, &user).await?;
    println!("Deleted user {}", user.user_name());

    Ok(())
}
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Rust.
  - AttachRolePolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolitica
  - DetachRolePolitica
  - PutUserPolitica

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea utenti IAM di sola lettura e lettura-scrittura utilizzando un SDK AWS

L'esempio di codice seguente mostra come creare degli utenti e collegarvi delle policy.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Creare due utenti IAM.
- Collega una policy che consenta a un utente di ottenere e inserire oggetti in un bucket Amazon S3.
- Collega una policy che consenta all'altro utente di ottenere oggetti dal bucket.
- Ottieni autorizzazioni diverse per il bucket in base alle credenziali dell'utente.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni degli utenti IAM.

```
import logging
import time

import boto3
from botocore.exceptions import ClientError

import access_key_wrapper
import policy_wrapper

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_user(user_name):
    """
    Creates a user. By default, a user has no permissions or access keys.

    :param user_name: The name of the user.
    :return: The newly created user.
    """
    try:
        user = iam.create_user(UserName=user_name)
        logger.info("Created user %s.", user.name)
    except ClientError:
        logger.exception("Couldn't create user %s.", user_name)
        raise
    else:
        return user



def update_user(user_name, new_user_name):
```

```
    """
    Updates a user's name.

    :param user_name: The current name of the user to update.
    :param new_user_name: The new name to assign to the user.
    :return: The updated user.
    """
    try:
        user = iam.User(user_name)
        user.update(NewUserName=new_user_name)
        logger.info("Renamed %s to %s.", user_name, new_user_name)
    except ClientError:
        logger.exception("Couldn't update name for user %s.", user_name)
        raise
    return user



def list_users():
    """
    Lists the users in the current account.

    :return: The list of users.
    """
    try:
        users = list(iam.users.all())
        logger.info("Got %s users.", len(users))
    except ClientError:
        logger.exception("Couldn't get users.")
        raise
    else:
        return users



def delete_user(user_name):
    """
    Deletes a user. Before a user can be deleted, all associated resources,
    such as access keys and policies, must be deleted or detached.

    :param user_name: The name of the user.
    """
    try:
        iam.User(user_name).delete()
```

```
        logger.info("Deleted user %s.", user_name)
    except ClientError:
        logger.exception("Couldn't delete user %s.", user_name)
        raise



def attach_policy(user_name, policy_arn):
    """
    Attaches a policy to a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to user %s.", policy_arn,
 user_name)
        raise



def detach_policy(user_name, policy_arn):
    """
    Detaches a policy from a user.

    :param user_name: The name of the user.
    :param policy_arn: The Amazon Resource Name (ARN) of the policy.
    """
    try:
        iam.User(user_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from user %s.", policy_arn, user_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from user %s.", policy_arn, user_name
        )
        raise
```

Crea funzioni che eseguono il wrapping delle operazioni delle policy IAM.

```
import json
import logging
import operator
import pprint
import time

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::my-bucket/*' to allow actions on all
 objects
                         in the bucket named 'my-bucket'.
    :return: The newly created policy.
    """
    policy_doc = {
        "Version": "2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
```

```
        raise
    else:
        return policy



def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
        raise
```

Crea funzioni che eseguono il wrapping delle operazioni delle chiavi di accesso IAM.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

iam = boto3.resource("iam")

def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
```

```
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
        raise
    else:
        return key_pair



def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

Utilizza le funzioni di wrapping per creare utenti con policy diverse e usa le loro credenziali per accedere a un bucket Amazon S3.

```
def usage_demo():
    """
    Shows how to manage users, keys, and policies.
    This demonstration creates two users: one user who can put and get objects in
 an
    Amazon S3 bucket, and another user who can only get objects from the bucket.
    The demo then shows how the users can perform only the actions they are
 permitted
    to perform.
```

```
    """
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management user demo.")
    print("-" * 88)
    print(
        "Users can have policies and roles attached to grant them specific "
        "permissions."
    )
    s3 = boto3.resource("s3")
    bucket = s3.create_bucket(
        Bucket=f"demo-iam-bucket-{time.time_ns()}",
        CreateBucketConfiguration={
            "LocationConstraint": s3.meta.client.meta.region_name
        },
    )
    print(f"Created an Amazon S3 bucket named {bucket.name}.")
    user_read_writer = create_user("demo-iam-read-writer")
    user_reader = create_user("demo-iam-reader")
    print(f"Created two IAM users: {user_read_writer.name} and
{user_reader.name}")
    update_user(user_read_writer.name, "demo-iam-creator")
    update_user(user_reader.name, "demo-iam-getter")
    users = list_users()
    user_read_writer = next(
        user for user in users if user.user_id == user_read_writer.user_id
    )
    user_reader = next(user for user in users if user.user_id ==
user_reader.user_id)
    print(
        f"Changed the names of the users to {user_read_writer.name} "
        f"and {user_reader.name}."
    )

    read_write_policy = policy_wrapper.create_policy(
        "demo-iam-read-write-policy",
        "Grants rights to create and get an object in the demo bucket.",
        ["s3:PutObject", "s3:GetObject"],
        f"arn:aws:s3:::{bucket.name}/*",
    )
    print(
        f"Created policy {read_write_policy.policy_name} with ARN:
{read_write_policy.arn}"
    )
```

```
    print(read_write_policy.description)
    read_policy = policy_wrapper.create_policy(
        "demo-iam-read-policy",
        "Grants rights to get an object from the demo bucket.",
        "s3:GetObject",
        f"arn:aws:s3:::{bucket.name}/*",
    )
    print(f"Created policy {read_policy.policy_name} with ARN:
{read_policy.arn}")
    print(read_policy.description)
    attach_policy(user_read_writer.name, read_write_policy.arn)
    print(f"Attached {read_write_policy.policy_name} to
{user_read_writer.name}.")
    attach_policy(user_reader.name, read_policy.arn)
    print(f"Attached {read_policy.policy_name} to {user_reader.name}.")

    user_read_writer_key = access_key_wrapper.create_key(user_read_writer.name)
    print(f"Created access key pair for {user_read_writer.name}.")
    user_reader_key = access_key_wrapper.create_key(user_reader.name)
    print(f"Created access key pair for {user_reader.name}.")

    s3_read_writer_resource = boto3.resource(
        "s3",
        aws_access_key_id=user_read_writer_key.id,
        aws_secret_access_key=user_read_writer_key.secret,
    )
    demo_object_key = f"object-{time.time_ns()}"
    demo_object = None
    while demo_object is None:
        try:
            demo_object = s3_read_writer_resource.Bucket(bucket.name).put_object(
                Key=demo_object_key, Body=b"AWS IAM demo object content!"
            )
        except ClientError as error:
            if error.response["Error"]["Code"] == "InvalidAccessKeyId":
                print("Access key not yet available. Waiting...")
                time.sleep(1)
            else:
                raise
    print(
        f"Put {demo_object_key} into {bucket.name} using "
        f"{user_read_writer.name}'s credentials."
    )
```

```
    read_writer_object = s3_read_writer_resource.Bucket(bucket.name).Object(
        demo_object_key
    )
    read_writer_content = read_writer_object.get()["Body"].read()
    print(f"Got object {read_writer_object.key} using read-writer user's
credentials.")
    print(f"Object content: {read_writer_content}")

    s3_reader_resource = boto3.resource(
        "s3",
        aws_access_key_id=user_reader_key.id,
        aws_secret_access_key=user_reader_key.secret,
    )
    demo_content = None
    while demo_content is None:
        try:
            demo_object =
s3_reader_resource.Bucket(bucket.name).Object(demo_object_key)
            demo_content = demo_object.get()["Body"].read()
            print(f"Got object {demo_object.key} using reader user's
credentials.")
            print(f"Object content: {demo_content}")
        except ClientError as error:
            if error.response["Error"]["Code"] == "InvalidAccessKeyId":
                print("Access key not yet available. Waiting...")
                time.sleep(1)
            else:
                raise

    try:
        demo_object.delete()
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("-" * 88)
            print(
                "Tried to delete the object using the reader user's credentials.
"
                "Got expected AccessDenied error because the reader is not "
                "allowed to delete objects."
            )
            print("-" * 88)

    access_key_wrapper.delete_key(user_reader.name, user_reader_key.id)
    detach_policy(user_reader.name, read_policy.arn)
```

```
    policy_wrapper.delete_policy(read_policy.arn)
    delete_user(user_reader.name)
    print(f"Deleted keys, detached and deleted policy, and deleted
{user_reader.name}.")

    access_key_wrapper.delete_key(user_read_writer.name, user_read_writer_key.id)
    detach_policy(user_read_writer.name, read_write_policy.arn)
    policy_wrapper.delete_policy(read_write_policy.arn)
    delete_user(user_read_writer.name)
    print(
        f"Deleted keys, detached and deleted policy, and deleted
{user_read_writer.name}."
    )

    bucket.objects.delete()
    bucket.delete()
    print(f"Emptied and deleted {bucket.name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AttachUserPolitica
  - CreateAccessChiave
  - CreatePolicy
  - CreateUser
  - DeleteAccessChiave
  - DeletePolicy
  - DeleteUser
  - DetachUserPolitica
  - ListUsers
  - UpdateUser

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci le chiavi di accesso IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come gestire le chiavi di accesso.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Creare ed elencare le chiavi di accesso.
- Scoprire come e quando una chiave di accesso è stata utilizzata per ultima.
- Aggiornare ed eliminare le chiavi di accesso.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni delle chiavi di accesso IAM.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

iam = boto3.resource("iam")

def list_keys(user_name):
    """
    Lists the keys owned by the specified user.

    :param user_name: The name of the user.
```

```
    :return: The list of keys owned by the user.
    """
    try:
        keys = list(iam.User(user_name).access_keys.all())
        logger.info("Got %s access keys for %s.", len(keys), user_name)
    except ClientError:
        logger.exception("Couldn't get access keys for %s.", user_name)
        raise
    else:
        return keys



def create_key(user_name):
    """
    Creates an access key for the specified user. Each user can have a
    maximum of two keys.

    :param user_name: The name of the user.
    :return: The created access key.
    """
    try:
        key_pair = iam.User(user_name).create_access_key_pair()
        logger.info(
            "Created access key pair for %s. Key ID is %s.",
            key_pair.user_name,
            key_pair.id,
        )
    except ClientError:
        logger.exception("Couldn't create access key pair for %s.", user_name)
        raise
    else:
        return key_pair



def get_last_use(key_id):
    """
    Gets information about when and how a key was last used.

    :param key_id: The ID of the key to look up.
    :return: Information about the key's last use.
    """
    try:
```

```
        response = iam.meta.client.get_access_key_last_used(AccessKeyId=key_id)
        last_used_date = response["AccessKeyLastUsed"].get("LastUsedDate", None)
        last_service = response["AccessKeyLastUsed"].get("ServiceName", None)
        logger.info(
            "Key %s was last used by %s on %s to access %s.",
            key_id,
            response["UserName"],
            last_used_date,
            last_service,
        )
    except ClientError:
        logger.exception("Couldn't get last use of key %s.", key_id)
        raise
    else:
        return response



def update_key(user_name, key_id, activate):
    """
    Updates the status of a key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to update.
    :param activate: When True, the key is activated. Otherwise, the key is
 deactivated.
    """

    try:
        key = iam.User(user_name).AccessKey(key_id)
        if activate:
            key.activate()
        else:
            key.deactivate()
        logger.info("%s key %s.", "Activated" if activate else "Deactivated",
 key_id)
    except ClientError:
        logger.exception(
            "Couldn't %s key %s.", "Activate" if activate else "Deactivate",
 key_id
        )
        raise
```

```
def delete_key(user_name, key_id):
    """
    Deletes a user's access key.

    :param user_name: The user that owns the key.
    :param key_id: The ID of the key to delete.
    """

    try:
        key = iam.AccessKey(user_name, key_id)
        key.delete()
        logger.info("Deleted access key %s for %s.", key.id, key.user_name)
    except ClientError:
        logger.exception("Couldn't delete key %s for %s", key_id, user_name)
        raise
```

Utilizza le funzioni di wrapping per eseguire operazioni sulle chiavi di accesso per l'utente corrente.

```
def usage_demo():
    """Shows how to create and manage access keys."""

    def print_keys():
        """Gets and prints the current keys for a user."""
        current_keys = list_keys(current_user_name)
        print("The current user's keys are now:")
        print(*[f"{key.id}: {key.status}" for key in current_keys], sep="\n")

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management access key demo.")
    print("-" * 88)
    current_user_name = iam.CurrentUser().user_name
    print(
        f"This demo creates an access key for the current user "
        f"({current_user_name}), manipulates the key in a few ways, and then "
        f"deletes it."
    )
    all_keys = list_keys(current_user_name)
```

```
    if len(all_keys) == 2:
        print(
            "The current user already has the maximum of 2 access keys. To run "
            "this demo, either delete one of the access keys or use a user "
            "that has only 1 access key."
        )
    else:
        new_key = create_key(current_user_name)
        print(f"Created a new key with id {new_key.id} and secret
{new_key.secret}.")
        print_keys()
        existing_key = next(key for key in all_keys if key != new_key)
        last_use = get_last_use(existing_key.id)["AccessKeyLastUsed"]
        print(
            f"Key {all_keys[0].id} was last used to access
{last_use['ServiceName']} "
            f"on {last_use['LastUsedDate']}"
        )
        update_key(current_user_name, new_key.id, False)
        print(f"Key {new_key.id} is now deactivated.")
        print_keys()
        delete_key(current_user_name, new_key.id)
        print_keys()
        print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CreateAccessChiave
  - DeleteAccessChiave
  - GetAccessKeyLastUsato
  - ListAccessChiavi
  - UpdateAccessChiave

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci le policy IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Creare ed elencare le policy.
- Creare ed ottenere le versioni della policy.
- Ripristinare una policy a una versione precedente.
- Eliminare le policy.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni delle policy IAM.

```
import json
import logging
import operator
import pprint
import time

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_policy(name, description, actions, resource_arn):
    """
    Creates a policy that contains a single statement.

    :param name: The name of the policy to create.
    :param description: The description of the policy.
    :param actions: The actions allowed by the policy. These typically take the
                    form of service:action, such as s3:PutObject.
```

```
    :param resource_arn: The Amazon Resource Name (ARN) of the resource this
 policy
                         applies to. This ARN can contain wildcards, such as
                         'arn:aws:s3:::my-bucket/*' to allow actions on all
 objects
                         in the bucket named 'my-bucket'.
    :return: The newly created policy.
    """
    policy_doc = {
        "Version": "2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.create_policy(
            PolicyName=name,
            Description=description,
            PolicyDocument=json.dumps(policy_doc),
        )
        logger.info("Created policy %s.", policy.arn)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy



def list_policies(scope):
    """
    Lists the policies in the current account.

    :param scope: Limits the kinds of policies that are returned. For example,
                  'Local' specifies that only locally managed policies are
 returned.
    :return: The list of policies.
    """
    try:
        policies = list(iam.policies.filter(Scope=scope))
        logger.info("Got %s policies in scope '%s'.", len(policies), scope)
    except ClientError:
        logger.exception("Couldn't get policies for scope '%s'.", scope)
        raise
    else:
```

```
        return policies



def create_policy_version(policy_arn, actions, resource_arn, set_as_default):
    """
    Creates a policy version. Policies can have up to five versions. The default
    version is the one that is used for all resources that reference the policy.

    :param policy_arn: The ARN of the policy.
    :param actions: The actions to allow in the policy version.
    :param resource_arn: The ARN of the resource this policy version applies to.
    :param set_as_default: When True, this policy version is set as the default
                           version for the policy. Otherwise, the default
                           is not changed.
    :return: The newly created policy version.
    """
    policy_doc = {
        "Version": "2012-10-17",
        "Statement": [{"Effect": "Allow", "Action": actions, "Resource":
 resource_arn}],
    }
    try:
        policy = iam.Policy(policy_arn)
        policy_version = policy.create_version(
            PolicyDocument=json.dumps(policy_doc), SetAsDefault=set_as_default
        )
        logger.info(
            "Created policy version %s for policy %s.",
            policy_version.version_id,
            policy_version.arn,
        )
    except ClientError:
        logger.exception("Couldn't create a policy version for %s.", policy_arn)
        raise
    else:
        return policy_version



def get_default_policy_statement(policy_arn):
    """
    Gets the statement of the default version of the specified policy.
```

```
    :param policy_arn: The ARN of the policy to look up.
    :return: The statement of the default policy version.
    """
    try:
        policy = iam.Policy(policy_arn)
        # To get an attribute of a policy, the SDK first calls get_policy.
        policy_doc = policy.default_version.document
        policy_statement = policy_doc.get("Statement", None)
        logger.info("Got default policy doc for %s.", policy.policy_name)
        logger.info(policy_doc)
    except ClientError:
        logger.exception("Couldn't get default policy statement for %s.",
 policy_arn)
        raise
    else:
        return policy_statement



def rollback_policy_version(policy_arn):
    """
    Rolls back to the previous default policy, if it exists.

    1. Gets the list of policy versions in order by date.
    2. Finds the default.
    3. Makes the previous policy the default.
    4. Deletes the old default version.

    :param policy_arn: The ARN of the policy to roll back.
    :return: The default version of the policy after the rollback.
    """
    try:
        policy_versions = sorted(
            iam.Policy(policy_arn).versions.all(),
            key=operator.attrgetter("create_date"),
        )
        logger.info("Got %s versions for %s.", len(policy_versions), policy_arn)
    except ClientError:
        logger.exception("Couldn't get versions for %s.", policy_arn)
        raise

    default_version = None
    rollback_version = None
    try:
```

```
        while default_version is None:
            ver = policy_versions.pop()
            if ver.is_default_version:
                default_version = ver
        rollback_version = policy_versions.pop()
        rollback_version.set_as_default()
        logger.info("Set %s as the default version.",
 rollback_version.version_id)
        default_version.delete()
        logger.info("Deleted original default version %s.",
 default_version.version_id)
    except IndexError:
        if default_version is None:
            logger.warning("No default version found for %s.", policy_arn)
        elif rollback_version is None:
            logger.warning(
                "Default version %s found for %s, but no previous version exists,
 so "
                "nothing to roll back to.",
                default_version.version_id,
                policy_arn,
            )
    except ClientError:
        logger.exception("Couldn't roll back version for %s.", policy_arn)
        raise
    else:
        return rollback_version



def delete_policy(policy_arn):
    """
    Deletes a policy.

    :param policy_arn: The ARN of the policy to delete.
    """
    try:
        iam.Policy(policy_arn).delete()
        logger.info("Deleted policy %s.", policy_arn)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_arn)
        raise
```

Utilizza le funzioni di wrapping per creare policy, aggiornare le versioni e ottenere informazioni su di esse.

```
def usage_demo():
    """Shows how to use the policy functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management policy demo.")
    print("-" * 88)
    print(
        "Policies let you define sets of permissions that can be attached to "
        "other IAM resources, like users and roles."
    )
    bucket_arn = f"arn:aws:s3:::made-up-bucket-name"
    policy = create_policy(
        "demo-iam-policy",
        "Policy for IAM demonstration.",
        ["s3:ListObjects"],
        bucket_arn,
    )
    print(f"Created policy {policy.policy_name}.")
    policies = list_policies("Local")
    print(f"Your account has {len(policies)} managed policies:")
    print(*[pol.policy_name for pol in policies], sep=", ")
    time.sleep(1)
    policy_version = create_policy_version(
        policy.arn, ["s3:PutObject"], bucket_arn, True
    )
    print(
        f"Added policy version {policy_version.version_id} to policy "
        f"{policy.policy_name}."
    )
    default_statement = get_default_policy_statement(policy.arn)
    print(f"The default policy statement for {policy.policy_name} is:")
    pprint.pprint(default_statement)
    rollback_version = rollback_policy_version(policy.arn)
    print(
        f"Rolled back to version {rollback_version.version_id} for "
        f"{policy.policy_name}."
    )
    default_statement = get_default_policy_statement(policy.arn)
```

```
    print(f"The default policy statement for {policy.policy_name} is now:")
    pprint.pprint(default_statement)
    delete_policy(policy.arn)
    print(f"Deleted policy {policy.policy_name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CreatePolicy
  - CreatePolicyVersione
  - DeletePolicy
  - DeletePolicyversione
  - GetPolicyversione
  - ListPolicies
  - ListPolicyVersioni
  - SetDefaultPolicyVersion

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci i ruoli IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM.
- Collegamento e scollegamento delle policy per un ruolo
- Elimina un ruolo.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni dei ruoli IAM.

```
import json
import logging
import pprint

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def create_role(role_name, allowed_services):
    """
    Creates a role that lets a list of specified services assume the role.

    :param role_name: The name of the role.
    :param allowed_services: The services that can assume the role.
    :return: The newly created role.
    """
    trust_policy = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {"Service": service},
                "Action": "sts:AssumeRole",
            }
            for service in allowed_services
        ],
    }

    try:
```

```
        role = iam.create_role(
            RoleName=role_name, AssumeRolePolicyDocument=json.dumps(trust_policy)
        )
        logger.info("Created role %s.", role.name)
    except ClientError:
        logger.exception("Couldn't create role %s.", role_name)
        raise
    else:
        return role



def attach_policy(role_name, policy_arn):
    """
    Attaches a policy to a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).attach_policy(PolicyArn=policy_arn)
        logger.info("Attached policy %s to role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception("Couldn't attach policy %s to role %s.", policy_arn,
 role_name)
        raise



def detach_policy(role_name, policy_arn):
    """
    Detaches a policy from a role.

    :param role_name: The name of the role. **Note** this is the name, not the
 ARN.
    :param policy_arn: The ARN of the policy.
    """
    try:
        iam.Role(role_name).detach_policy(PolicyArn=policy_arn)
        logger.info("Detached policy %s from role %s.", policy_arn, role_name)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from role %s.", policy_arn, role_name
```

```
        )
        raise



def delete_role(role_name):
    """
    Deletes a role.

    :param role_name: The name of the role to delete.
    """
    try:
        iam.Role(role_name).delete()
        logger.info("Deleted role %s.", role_name)
    except ClientError:
        logger.exception("Couldn't delete role %s.", role_name)
        raise
```

Utilizza le funzioni di wrapping per creare un ruolo, per poi collegare e scollegare una policy.

```
def usage_demo():
    """Shows how to use the role functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management role demo.")
    print("-" * 88)
    print(
        "Roles let you define sets of permissions and can be assumed by "
        "other entities, like users and services."
    )
    print("The first 10 roles currently in your account are:")
    roles = list_roles(10)
    print(f"The inline policies for role {roles[0].name} are:")
    list_policies(roles[0].name)
    role = create_role(
        "demo-iam-role", ["lambda.amazonaws.com",
 "batchoperations.s3.amazonaws.com"]
    )
    print(f"Created role {role.name}, with trust policy:")
    pprint.pprint(role.assume_role_policy_document)
```

```
    policy_arn = "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"
    attach_policy(role.name, policy_arn)
    print(f"Attached policy {policy_arn} to {role.name}.")
    print(f"Policies attached to role {role.name} are:")
    list_attached_policies(role.name)
    detach_policy(role.name, policy_arn)
    print(f"Detached policy {policy_arn} from {role.name}.")
    delete_role(role.name)
    print(f"Deleted {role.name}.")
    print("Thanks for watching!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AttachRolePolitica
  - CreateRole
  - DeleteRole
  - DetachRolePolitica

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci il tuo account IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Ottenere e aggiornare l'alias dell'account.
- Generare un report degli utenti e delle loro credenziali.
- Ottenere un riepilogo dell'utilizzo dell'account.
- Ottenere informazioni su tutti gli utenti, gruppi, ruoli e policy nell'account, comprese le relazioni reciproche.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrapping delle operazioni dell'account IAM.

```
import logging
import pprint
import sys
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
iam = boto3.resource("iam")

def list_aliases():
    """
    Gets the list of aliases for the current account. An account has at most one
 alias.

    :return: The list of aliases for the account.
    """
    try:
        response = iam.meta.client.list_account_aliases()
        aliases = response["AccountAliases"]
        if len(aliases) > 0:
            logger.info("Got aliases for your account: %s.", ",".join(aliases))
        else:
            logger.info("Got no aliases for your account.")
    except ClientError:
        logger.exception("Couldn't list aliases for your account.")
        raise
    else:
        return response["AccountAliases"]
```

```
def create_alias(alias):
    """
    Creates an alias for the current account. The alias can be used in place of
 the
    account ID in the sign-in URL. An account can have only one alias. When a new
    alias is created, it replaces any existing alias.

    :param alias: The alias to assign to the account.
    """

    try:
        iam.create_account_alias(AccountAlias=alias)
        logger.info("Created an alias '%s' for your account.", alias)
    except ClientError:
        logger.exception("Couldn't create alias '%s' for your account.", alias)
        raise



def delete_alias(alias):
    """
    Removes the alias from the current account.

    :param alias: The alias to remove.
    """
    try:
        iam.meta.client.delete_account_alias(AccountAlias=alias)
        logger.info("Removed alias '%s' from your account.", alias)
    except ClientError:
        logger.exception("Couldn't remove alias '%s' from your account.", alias)
        raise



def generate_credential_report():
    """
    Starts generation of a credentials report about the current account. After
    calling this function to generate the report, call get_credential_report
    to get the latest report. A new report can be generated a minimum of four
 hours
    after the last one was generated.
    """
    try:
```

```
        response = iam.meta.client.generate_credential_report()
        logger.info(
            "Generating credentials report for your account. " "Current state is
 %s.",
            response["State"],
        )
    except ClientError:
        logger.exception("Couldn't generate a credentials report for your
 account.")
        raise
    else:
        return response



def get_credential_report():
    """
    Gets the most recently generated credentials report about the current
 account.

    :return: The credentials report.
    """
    try:
        response = iam.meta.client.get_credential_report()
        logger.debug(response["Content"])
    except ClientError:
        logger.exception("Couldn't get credentials report.")
        raise
    else:
        return response["Content"]



def get_summary():
    """
    Gets a summary of account usage.

    :return: The summary of account usage.
    """
    try:
        summary = iam.AccountSummary()
        logger.debug(summary.summary_map)
    except ClientError:
        logger.exception("Couldn't get a summary for your account.")
```

```
        raise
    else:
        return summary.summary_map



def get_authorization_details(response_filter):
    """
    Gets an authorization detail report for the current account.

    :param response_filter: A list of resource types to include in the report,
 such
                            as users or roles. When not specified, all resources
                            are included.
    :return: The authorization detail report.
    """
    try:
        account_details = iam.meta.client.get_account_authorization_details(
            Filter=response_filter
        )
        logger.debug(account_details)
    except ClientError:
        logger.exception("Couldn't get details for your account.")
        raise
    else:
        return account_details
```

Utilizza le funzioni di wrapping per modificare l'alias dell'account e recuperare report sull'account.

```
def usage_demo():
    """Shows how to use the account functions."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    print("-" * 88)
    print("Welcome to the AWS Identity and Account Management account demo.")
    print("-" * 88)
    print(
        "Setting an account alias lets you use the alias in your sign-in URL "
        "instead of your account number."
    )
```

```
    old_aliases = list_aliases()
    if len(old_aliases) > 0:
        print(f"Your account currently uses '{old_aliases[0]}' as its alias.")
    else:
        print("Your account currently has no alias.")
    for index in range(1, 3):
        new_alias = f"alias-{index}-{time.time_ns()}"
        print(f"Setting your account alias to {new_alias}")
        create_alias(new_alias)
    current_aliases = list_aliases()
    print(f"Your account alias is now {current_aliases}.")
    delete_alias(current_aliases[0])
    print(f"Your account now has no alias.")
    if len(old_aliases) > 0:
        print(f"Restoring your original alias back to {old_aliases[0]}...")
        create_alias(old_aliases[0])

    print("-" * 88)
    print("You can get various reports about your account.")
    print("Let's generate a credentials report...")
    report_state = None
    while report_state != "COMPLETE":
        cred_report_response = generate_credential_report()
        old_report_state = report_state
        report_state = cred_report_response["State"]
        if report_state != old_report_state:
            print(report_state, sep="")
        else:
            print(".", sep="")
        sys.stdout.flush()
        time.sleep(1)
    print()
    cred_report = get_credential_report()
    col_count = 3
    print(f"Got credentials report. Showing only the first {col_count} columns.")
    cred_lines = [
        line.split(",")[:col_count] for line in
 cred_report.decode("utf-8").split("\n")
    ]
    col_width = max([len(item) for line in cred_lines for item in line]) + 2
    for line in cred_report.decode("utf-8").split("\n"):
        print(
            "".join(element.ljust(col_width) for element in line.split(",")
[:col_count])
```

```
        )

    print("-" * 88)
    print("Let's get an account summary.")
    summary = get_summary()
    print("Here's your summary:")
    pprint.pprint(summary)

    print("-" * 88)
    print("Let's get authorization details!")
    details = get_authorization_details([])
    see_details = input("These are pretty long, do you want to see them (y/n)? ")
    if see_details.lower() == "y":
        pprint.pprint(details)

    print("-" * 88)
    pw_policy_created = None
    see_pw_policy = input("Want to see the password policy for the account (y/n)?
")
    if see_pw_policy.lower() == "y":
        while True:
            if print_password_policy():
                break
            else:
                answer = input(
                    "Do you want to create a default password policy (y/n)? "
                )
                if answer.lower() == "y":
                    pw_policy_created = iam.create_account_password_policy()
                else:
                    break
    if pw_policy_created is not None:
        answer = input("Do you want to delete the password policy (y/n)? ")
        if answer.lower() == "y":
            pw_policy_created.delete()
            print("Password policy deleted.")

    print("The SAML providers for your account are:")
    list_saml_providers(10)

    print("-" * 88)
    print("Thanks for watching.")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CreateAccountPseudonimo
  - DeleteAccountPseudonimo
  - GenerateCredentialRapporto
  - GetAccountAuthorizationDetails
  - GetAccountRiepilogo
  - GetCredentialRapporto
  - ListAccountPseudonimi

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ripristina una versione della policy IAM utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Ottieni l'elenco delle versioni delle policy in ordine di data.
- Individua la versione predefinita della policy.
- Rendi predefinita la versione precedente della policy.
- Elimina la vecchia versione predefinita.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
def rollback_policy_version(policy_arn):
```

```
    """
    Rolls back to the previous default policy, if it exists.

    1. Gets the list of policy versions in order by date.
    2. Finds the default.
    3. Makes the previous policy the default.
    4. Deletes the old default version.

    :param policy_arn: The ARN of the policy to roll back.
    :return: The default version of the policy after the rollback.
    """
    try:
        policy_versions = sorted(
            iam.Policy(policy_arn).versions.all(),
            key=operator.attrgetter("create_date"),
        )
        logger.info("Got %s versions for %s.", len(policy_versions), policy_arn)
    except ClientError:
        logger.exception("Couldn't get versions for %s.", policy_arn)
        raise

    default_version = None
    rollback_version = None
    try:
        while default_version is None:
            ver = policy_versions.pop()
            if ver.is_default_version:
                default_version = ver
        rollback_version = policy_versions.pop()
        rollback_version.set_as_default()
        logger.info("Set %s as the default version.",
rollback_version.version_id)
        default_version.delete()
        logger.info("Deleted original default version %s.",
default_version.version_id)
    except IndexError:
        if default_version is None:
            logger.warning("No default version found for %s.", policy_arn)
        elif rollback_version is None:
            logger.warning(
                "Default version %s found for %s, but no previous version exists,
so "
                "nothing to roll back to.",
                default_version.version_id,
```

```
                policy_arn,
            )
    except ClientError:
        logger.exception("Couldn't roll back version for %s.", policy_arn)
        raise
    else:
        return rollback_version
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - DeletePolicyVersione
  - ListPolicyVersioni
  - SetDefaultPolicyVersion

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Lavora con l'API IAM Policy Builder utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Crea policy IAM utilizzando l'API orientata agli oggetti.
- Usa l'API IAM Policy Builder con il servizio IAM.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Gli esempi utilizzano le seguenti importazioni.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.policybuilder.iam.IamConditionOperator;
import software.amazon.awssdk.policybuilder.iam.IamEffect;
import software.amazon.awssdk.policybuilder.iam.IamPolicy;
import software.amazon.awssdk.policybuilder.iam.IamPolicyWriter;
import software.amazon.awssdk.policybuilder.iam.IamPrincipal;
import software.amazon.awssdk.policybuilder.iam.IamPrincipalType;
import software.amazon.awssdk.policybuilder.iam.IamResource;
import software.amazon.awssdk.policybuilder.iam.IamStatement;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.iam.IamClient;
import software.amazon.awssdk.services.iam.model.GetPolicyResponse;
import software.amazon.awssdk.services.iam.model.GetPolicyVersionResponse;
import software.amazon.awssdk.services.sts.StsClient;

import java.net.URLDecoder;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.List;
```

Crea una policy basata sul tempo.

```
        public String timeBasedPolicyExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:GetItem")
                                                .addResource(IamResource.ALL)
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.DATE_GREATER_THAN)

 .key("aws:CurrentTime")

 .value("2020-04-01T00:00:00Z"))
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.DATE_LESS_THAN)

 .key("aws:CurrentTime")
```

```
 .value("2020-06-30T23:59:59Z")))
                                .build();

                // Use an IamPolicyWriter to write out the JSON string to a more
 readable
                // format.
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true)
                                .build());
        }
```

Crea una policy con più condizioni.

```
        public String multipleConditionsExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:GetItem")

 .addAction("dynamodb:BatchGetItem")
                                                .addAction("dynamodb:Query")
                                                .addAction("dynamodb:PutItem")
                                                .addAction("dynamodb:UpdateItem")
                                                .addAction("dynamodb:DeleteItem")

 .addAction("dynamodb:BatchWriteItem")

 .addResource("arn:aws:dynamodb:*:*:table/table-name")

 .addConditions(IamConditionOperator.STRING_EQUALS

 .addPrefix("ForAllValues:"),

 "dynamodb:Attributes",
                                                                        List.of("column-
name1", "column-name2", "column-name3"))
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.STRING_EQUALS

 .addSuffix("IfExists"))
```

```
.key("dynamodb:Select")

.value("SPECIFIC_ATTRIBUTES")))
                                .build();

                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Usa i principi in una policy.

```
        public String specifyPrincipalsExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.DENY)
                                                .addAction("s3:*")
                                                .addPrincipal(IamPrincipal.ALL)

.addResource("arn:aws:s3:::BUCKETNAME/*")

.addResource("arn:aws:s3:::BUCKETNAME")
                                                .addCondition(b1 -> b1

.operator(IamConditionOperator.ARN_NOT_EQUALS)

.key("aws:PrincipalArn")

.value("arn:aws:iam::444455556666:user/user-name")))
                                .build();
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Consentire l'accesso multi-account .

```
        public String allowCrossAccountAccessExample() {
                IamPolicy policy = IamPolicy.builder()
                                .addStatement(b -> b
                                                .effect(IamEffect.ALLOW)
```

```
 .addPrincipal(IamPrincipalType.AWS, "111122223333")
                                                .addAction("s3:PutObject")
                                                .addResource("arn:aws:s3:::DOC-
EXAMPLE-BUCKET/*")
                                                .addCondition(b1 -> b1

 .operator(IamConditionOperator.STRING_EQUALS)
                                                                    .key("s3:x-amz-
acl")
                                                                    .value("bucket-
owner-full-control")))
                                .build();
                return policy.toJson(IamPolicyWriter.builder()
                                .prettyPrint(true).build());
        }
```

Crea e carica un `IamPolicy`.

```
        public String createAndUploadPolicyExample(IamClient iam, String
 accountID, String policyName) {
                // Build the policy.
                IamPolicy policy = IamPolicy.builder() // 'version' defaults to
 "2012-10-17".
                                .addStatement(IamStatement.builder()
                                                .effect(IamEffect.ALLOW)
                                                .addAction("dynamodb:PutItem")

 .addResource("arn:aws:dynamodb:us-east-1:" + accountID
                                                                    + ":table/
exampleTableName")
                                                .build())
                                .build();
                // Upload the policy.
                iam.createPolicy(r ->
 r.policyName(policyName).policyDocument(policy.toJson()));
                return
 policy.toJson(IamPolicyWriter.builder().prettyPrint(true).build());
        }
```

Scarica e lavora con un `IamPolicy`.

```
       public String createNewBasedOnExistingPolicyExample(IamClient iam, String
accountID, String policyName,
                       String newPolicyName) {

               String policyArn = "arn:aws:iam::" + accountID + ":policy/" +
policyName;
               GetPolicyResponse getPolicyResponse = iam.getPolicy(r ->
r.policyArn(policyArn));

               String policyVersion =
getPolicyResponse.policy().defaultVersionId();
               GetPolicyVersionResponse getPolicyVersionResponse = iam
                               .getPolicyVersion(r ->
r.policyArn(policyArn).versionId(policyVersion));

               // Create an IamPolicy instance from the JSON string returned
from IAM.
               String decodedPolicy =
URLDecoder.decode(getPolicyVersionResponse.policyVersion().document(),
                               StandardCharsets.UTF_8);
               IamPolicy policy = IamPolicy.fromJson(decodedPolicy);

               /*
                * All IamPolicy components are immutable, so use the copy method
that creates a
                * new instance that
                * can be altered in the same method call.
                *
                * Add the ability to get an item from DynamoDB as an additional
action.
                */
               IamStatement newStatement = policy.statements().get(0).copy(s ->
s.addAction("dynamodb:GetItem"));

               // Create a new statement that replaces the original statement.
               IamPolicy newPolicy = policy.copy(p ->
p.statements(Arrays.asList(newStatement)));

               // Upload the new policy. IAM now has both policies.
               iam.createPolicy(r -> r.policyName(newPolicyName)
                               .policyDocument(newPolicy.toJson()));
```

```
                return
newPolicy.toJson(IamPolicyWriter.builder().prettyPrint(true).build());
        }
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for Java 2.x.
- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - CreatePolicy
  - GetPolicy
  - GetPolicyVersione

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di IAM con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice per l' AWS STS utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come utilizzare un kit AWS STS di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Azioni per l' AWS STS utilizzo degli AWS SDK
  - Utilizzo AssumeRole con un AWS SDK o una CLI
  - Utilizzo AssumeRoleWithWebIdentity con un AWS SDK o una CLI
  - Utilizzo DecodeAuthorizationMessage con un AWS SDK o una CLI

- Utilizzo GetFederationToken con un AWS SDK o una CLI
- Utilizzo GetSessionToken con un AWS SDK o una CLI

- Scenari per l' AWS STS utilizzo degli AWS SDK
  - Assumi un ruolo IAM che richiede un token MFA con l' AWS STS utilizzo di un SDK AWS
  - Costruisci un URL con AWS STS per utenti federati utilizzando un SDK AWS
  - Ottieni un token di sessione che richiede un token MFA AWS STS utilizzando un SDK AWS

Azioni per l' AWS STS utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come eseguire AWS STS azioni individuali con gli AWS SDK. Questi estratti richiamano l' AWS STS API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la Documentazione di riferimento delle API AWS Security Token Service (AWS STS).

Esempi

- Utilizzo AssumeRole con un AWS SDK o una CLI
- Utilizzo AssumeRoleWithWebIdentity con un AWS SDK o una CLI
- Utilizzo DecodeAuthorizationMessage con un AWS SDK o una CLI
- Utilizzo GetFederationToken con un AWS SDK o una CLI
- Utilizzo GetSessionToken con un AWS SDK o una CLI

Utilizzo **`AssumeRole`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AssumeRole`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Assunzione un ruolo IAM che richiede un token MFA
- Formulazione di un URL per gli utenti federati

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;

namespace AssumeRoleExample
{
    class AssumeRole
    {
        /// <summary>
        /// This example shows how to use the AWS Security Token
        /// Service (AWS STS) to assume an IAM role.
        ///
        /// NOTE: It is important that the role that will be assumed has a
        /// trust relationship with the account that will assume the role.
        ///
        /// Before you run the example, you need to create the role you want to
        /// assume and have it trust the IAM account that will assume that role.
        ///
        /// See https://docs.aws.amazon.com/IAM/latest/UserGuide/
id_roles_create.html
        /// for help in working with roles.
        /// </summary>

        private static readonly RegionEndpoint REGION = RegionEndpoint.USWest2;

        static async Task Main()
        {
            // Create the SecurityToken client and then display the identity of
 the
            // default user.
```

```
            var roleArnToAssume = "arn:aws:iam::123456789012:role/
testAssumeRole";

            var client = new
 Amazon.SecurityToken.AmazonSecurityTokenServiceClient(REGION);

            // Get and display the information about the identity of the default
 user.
            var callerIdRequest = new GetCallerIdentityRequest();
            var caller = await client.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"Original Caller: {caller.Arn}");

            // Create the request to use with the AssumeRoleAsync call.
            var assumeRoleReq = new AssumeRoleRequest()
            {
                DurationSeconds = 1600,
                RoleSessionName = "Session1",
                RoleArn = roleArnToAssume
            };

            var assumeRoleRes = await client.AssumeRoleAsync(assumeRoleReq);

            // Now create a new client based on the credentials of the caller
 assuming the role.
            var client2 = new AmazonSecurityTokenServiceClient(credentials:
 assumeRoleRes.Credentials);

            // Get and display information about the caller that has assumed the
 defined role.
            var caller2 = await client2.GetCallerIdentityAsync(callerIdRequest);
            Console.WriteLine($"AssumedRole Caller: {caller2.Arn}");
        }
    }
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
##############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
##############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
#  credentials.
#
# Parameters:
#       -n role_session_name -- The name of the session.
#       -r role_arn -- The ARN of the role to assume.
#
# Returns:
```

```
#       [access_key_id, secret_access_key, session_token]
#     And:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function sts_assume_role() {
  local role_session_name role_arn response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary
 credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
  }

  while getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        ech o"Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
```

```
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- Per i dettagli sull'API, consulta AssumeRole AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::STS::assumeRole(const Aws::String &roleArn,
                             const Aws::String &roleSessionName,
                             const Aws::String &externalId,
                             Aws::Auth::AWSCredentials &credentials,
                             const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::STS::STSClient sts(clientConfig);
    Aws::STS::Model::AssumeRoleRequest sts_req;

    sts_req.SetRoleArn(roleArn);
    sts_req.SetRoleSessionName(roleSessionName);
    sts_req.SetExternalId(externalId);

    const Aws::STS::Model::AssumeRoleOutcome outcome = sts.AssumeRole(sts_req);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error assuming IAM role. " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
```

```
    else {
        std::cout << "Credentials successfully retrieved." << std::endl;
        const Aws::STS::Model::AssumeRoleResult result = outcome.GetResult();
        const Aws::STS::Model::Credentials &temp_credentials =
 result.GetCredentials();

        // Store temporary credentials in return argument.
        // Note: The credentials object returned by assumeRole differs
        // from the AWSCredentials object used in most situations.
        credentials.SetAWSAccessKeyId(temp_credentials.GetAccessKeyId());
        credentials.SetAWSSecretKey(temp_credentials.GetSecretAccessKey());
        credentials.SetSessionToken(temp_credentials.GetSessionToken());
    }

    return outcome.IsSuccess();
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Come assumere un ruolo

Il comando `assume-role` seguente recupera un set di credenziali a breve termine per il ruolo IAM `s3-access-example`.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

Output:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-
access-example"
    },
    "Credentials": {
```

```
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8B
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che puoi utilizzare per l'autenticazione in AWS.

Per l'utilizzo della AWS CLI, è possibile impostare un profilo denominato associato a un ruolo. Quando utilizzi il profilo, la AWS CLI chiamerà assume-role e gestirà le credenziali per te. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM nella CLI nella AWS CLI User Guide AWS .

- Per i dettagli sull'API, consulta AWS CLI Command AssumeRoleReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sts.StsClient;
import software.amazon.awssdk.services.sts.model.AssumeRoleRequest;
import software.amazon.awssdk.services.sts.model.StsException;
import software.amazon.awssdk.services.sts.model.AssumeRoleResponse;
import software.amazon.awssdk.services.sts.model.Credentials;
import java.time.Instant;
import java.time.ZoneId;
```

```
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;
import java.util.Locale;

/**
 * To make this code example work, create a Role that you want to assume.
 * Then define a Trust Relationship in the AWS Console. You can use this as an
 * example:
 *
 * {
 * "Version": "2012-10-17",
 * "Statement": [
 * {
 * "Effect": "Allow",
 * "Principal": {
 * "AWS": "<Specify the ARN of your IAM user you are using in this code
 * example>"
 * },
 * "Action": "sts:AssumeRole"
 * }
 * ]
 * }
 *
 * For more information, see "Editing the Trust Relationship for an Existing
 * Role" in the AWS Directory Service guide.
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class AssumeRole {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <roleArn> <roleSessionName>\s

                Where:
                    roleArn - The Amazon Resource Name (ARN) of the role to
 assume (for example, rn:aws:iam::000008047983:role/s3role).\s
```

```
                    roleSessionName - An identifier for the assumed role session
(for example, mysession).\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String roleArn = args[0];
        String roleSessionName = args[1];
        Region region = Region.US_EAST_1;
        StsClient stsClient = StsClient.builder()
                .region(region)
                .build();

        assumeGivenRole(stsClient, roleArn, roleSessionName);
        stsClient.close();
    }

    public static void assumeGivenRole(StsClient stsClient, String roleArn,
String roleSessionName) {
        try {
            AssumeRoleRequest roleRequest = AssumeRoleRequest.builder()
                    .roleArn(roleArn)
                    .roleSessionName(roleSessionName)
                    .build();

            AssumeRoleResponse roleResponse = stsClient.assumeRole(roleRequest);
            Credentials myCreds = roleResponse.credentials();

            // Display the time when the temp creds expire.
            Instant exTime = myCreds.expiration();
            String tokenInfo = myCreds.sessionToken();

            // Convert the Instant to readable date.
            DateTimeFormatter formatter =
DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
                    .withLocale(Locale.US)
                    .withZone(ZoneId.systemDefault());

            formatter.format(exTime);
            System.out.println("The token " + tokenInfo + "  expires on " +
exTime);
```

```
        } catch (StsException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea il client.

```
import { STSClient } from "@aws-sdk/client-sts";
// Set the AWS Region.
const REGION = "us-east-1";
// Create an AWS STS service client object.
export const client = new STSClient({ region: REGION });
```

Assumi il ruolo IAM.

```
import { AssumeRoleCommand } from "@aws-sdk/client-sts";

import { client } from "../libs/client.js";

export const main = async () => {
  try {
    // Returns a set of temporary security credentials that you can use to
    // access Amazon Web Services resources that you might not normally
    // have access to.
```

```
    const command = new AssumeRoleCommand({
      // The Amazon Resource Name (ARN) of the role to assume.
      RoleArn: "ROLE_ARN",
      // An identifier for the assumed role session.
      RoleSessionName: "session1",
      // The duration, in seconds, of the role session. The value specified
      // can range from 900 seconds (15 minutes) up to the maximum session
      // duration set for the role.
      DurationSeconds: 900,
    });
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
const AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

var roleToAssume = {
  RoleArn: "arn:aws:iam::123456789012:role/RoleName",
  RoleSessionName: "session1",
  DurationSeconds: 900,
};
var roleCreds;

// Create the STS service object
var sts = new AWS.STS({ apiVersion: "2011-06-15" });
```

```
//Assume Role
sts.assumeRole(roleToAssume, function (err, data) {
  if (err) console.log(err, err.stack);
  else {
    roleCreds = {
      accessKeyId: data.Credentials.AccessKeyId,
      secretAccessKey: data.Credentials.SecretAccessKey,
      sessionToken: data.Credentials.SessionToken,
    };
    stsGetCallerIdentity(roleCreds);
  }
});

//Get Arn of current identity
function stsGetCallerIdentity(creds) {
  var stsParams = { credentials: creds };
  // Create STS service object
  var sts = new AWS.STS(stsParams);

  sts.getCallerIdentity({}, function (err, data) {
    if (err) {
      console.log(err, err.stack);
    } else {
      console.log(data.Arn);
    }
  });
}
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce un set di credenziali temporanee (chiave di accesso, chiave segreta e token di sessione) che possono essere utilizzate per un'ora per accedere a AWS risorse a cui l'utente richiedente potrebbe normalmente non avere accesso. Le credenziali restituite hanno le autorizzazioni consentite dalla politica di accesso del ruolo assunto e dalla politica fornita (non è possibile utilizzare la politica fornita per concedere autorizzazioni superiori a quelle definite dalla politica di accesso del ruolo assunto).

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -Policy "...JSON policy..." -DurationInSeconds 3600
```

Esempio 2: restituisce un set di credenziali temporanee, valide per un'ora, con le stesse autorizzazioni definite nella politica di accesso del ruolo assunto.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600
```

Esempio 3: restituisce un set di credenziali temporanee che forniscono il numero di serie e il token generato da un MFA associato alle credenziali utente utilizzate per eseguire il cmdlet.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -SerialNumber "GAHT12345678" -TokenCode "123456"
```

Esempio 4: restituisce un set di credenziali temporanee che hanno assunto un ruolo definito in un account cliente. Per ogni ruolo che la terza parte può assumere, l'account cliente deve creare un ruolo utilizzando un identificatore che deve essere passato nel ExternalId parametro - ogni volta che viene assunto il ruolo.

```
Use-STSRole -RoleSessionName "Bob" -RoleArn "arn:aws:iam::123456789012:role/demo"
 -DurationInSeconds 3600 -ExternalId "ABC123"
```

- Per i dettagli sull'API, vedere AssumeRolein AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Assumi un ruolo IAM che richiede un token MFA e utilizza le credenziali temporanee per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_from_assumed_role_with_mfa(
```

```
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an ARN.
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
 # Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these
credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to
assume the role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: "create-use-assume-role-scenario"
   )
   @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
 end
```

- Per i dettagli sull'API, AssumeRoleconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
async fn assume_role(config: &SdkConfig, role_name: String, session_name:
 Option<String>) {
    let provider = aws_config::sts::AssumeRoleProvider::builder(role_name)
        .session_name(session_name.unwrap_or("rust_sdk_example_session".into()))
        .configure(config)
        .build()
        .await;

    let local_config = aws_config::from_env()
        .credentials_provider(provider)
        .load()
        .await;
    let client = Client::new(&local_config);
    let req = client.get_caller_identity();
    let resp = req.send().await;
    match resp {
        Ok(e) => {
            println!("UserID :               {}",
 e.user_id().unwrap_or_default());
            println!("Account:               {}",
 e.account().unwrap_or_default());
            println!("Arn    :               {}", e.arn().unwrap_or_default());
        }
        Err(e) => println!("{:?}", e),
    }
}
```

- Per i dettagli sulle API, consulta il riferimento AssumeRoleall'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
public func assumeRole(role: IAMClientTypes.Role, sessionName: String)
                async throws -> STSClientTypes.Credentials {
    let input = AssumeRoleInput(
        roleArn: role.arn,
        roleSessionName: sessionName
    )
    do {
        let output = try await stsClient.assumeRole(input: input)

        guard let credentials = output.credentials else {
            throw ServiceHandlerError.authError
        }

        return credentials
    } catch {
        throw error
    }
}
```

- Per i dettagli sull'API, consulta la AssumeRoleguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AssumeRoleWithWebIdentity`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AssumeRoleWithWebIdentity`.

CLI

AWS CLI

Per ottenere credenziali a breve termine per un ruolo autenticato con Web Identity (OAuth 2."0)

Il comando `assume-role-with-web-identity` seguente recupera un set di credenziali a breve termine per il ruolo IAM `app1`. La richiesta viene autenticata utilizzando il token di identità Web fornito dal provider di identità Web specificato. Alla sessione vengono applicate due politiche aggiuntive per limitare ulteriormente ciò che l'utente può fare. Le credenziali restituite scadono un'ora dopo la generazione.

```
aws sts assume-role-with-web-identity \
    --duration-seconds 3600 \
    --role-session-name "app1" \
    --provider-id "www.amazon.com" \
    --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/
webidentitydemopolicy2" \
    --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
    --web-identity-token "Atza
%7CIQEBLjAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCz
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wl7WTI7jn-Pcb6M-
buCgHhFOzTQxod27L9CqnOLio7N3gZAGpsp6n1-
AJBOCJckcyXe2c6uD0srOJeZlKUm2eTDVMf8IehDVI0r1QOnTV6KzzAI3OY87Vd_cVMQ"
```

Output:

```
{
    "SubjectFromWebIdentityToken": "amzn1.account.AF6RHO7KZU5XRVQJGXK6HB56KR2A"
    "Audience": "client.5498841531868486423.1548@apps.example.com",
    "AssumedRoleUser": {
```

```
        "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
        "AssumedRoleId": "AROACLKWSDQRAOEXAMPLE:app1"
    }
    "Credentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    },
    "Provider": "www.amazon.com"
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta AssumeRoleWithWebIdentity in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce un set temporaneo di credenziali, valido per un'ora, per un utente che è stato autenticato con il provider di identità Login with Amazon. Le credenziali presuppongono la politica di accesso associata al ruolo identificato dal ruolo ARN. Facoltativamente, è possibile passare una policy JSON al parametro -Policy che perfeziona ulteriormente le autorizzazioni di accesso (non è possibile concedere più autorizzazioni di quelle disponibili nelle autorizzazioni associate al ruolo). Il valore fornito a - WebIdentityToken è l'identificatore utente univoco restituito dal provider di identità.

```
Use-STSWebIdentityRole -DurationInSeconds 3600 -ProviderId "www.amazon.com"
 -RoleSessionName "app1" -RoleArn "arn:aws:iam::123456789012:role/
FederatedWebIdentityRole" -WebIdentityToken "Atza...DVI0r1"
```

- Per i dettagli sull'API, vedere AssumeRoleWithWebIdentity in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DecodeAuthorizationMessage`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DecodeAuthorizationMessage`.

CLI

AWS CLI

Per decodificare un messaggio di autorizzazione codificato restituito in risposta a una richiesta

L'`decode-authorization-message`esempio seguente decodifica informazioni aggiuntive sullo stato di autorizzazione di una richiesta da un messaggio codificato restituito in risposta a una richiesta Amazon Web Services.

```
aws sts decode-authorization-message \
    --encoded-message EXAMPLEWodyRNrtlQARDip-
eTA6i6DrlUhHhPQrLWB_lAbl5pAKxl9mPDLexYcGBreyIKQC1BGBIpBKr3dFDkwqeO7e2NMk5j_hmzAiChJN-8oy3
Ojau7BMjOTWwOtHPhV_Zaz87yENdipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK
pOu3FZjwYStfvTb3GHs3-6rLribGO9jZOktkfE6vqxlFzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgo
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHlI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3fOHanFuBZgENxZQZ2dlH3xJGMTtYayzZrRXjiq_SfX9
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjjOtNPBOZCcuEKE
```

Output:

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,
\"matchedStatements\":{\"items\":[{\"statementId\":\"VisualEditor0\",\"effect
\":\"DENY\",\"principals\":{\"items\":[{\"value\":\"AROA123456789EXAMPLE
\"}]},\"principalGroups\":{\"items\":[]},\"actions\":{\"items\":[{\"value
\":\"ec2:RunInstances\"}]},\"resources\":{\"items\":[{\"value\":\"*
\"}]},\"conditions\":{\"items\":[]}}]},\"failures\":{\"items\":[]},
\"context\":{\"principal\":{\"id\":\"AROA123456789EXAMPLE:Ana\",\"arn
\":\"arn:aws:sts::111122223333:assumed-role/Developer/Ana\"},\"action\":
\"RunInstances\",\"resource\":\"arn:aws:ec2:us-east-1:111122223333:instance/*
\",\"conditions\":{\"items\":[{\"key\":\"ec2:MetadataHttpPutResponseHopLimit\",
\"values\":{\"items\":[{\"value\":\"2\"}]}},{\"key\":\"ec2:InstanceMarketType
\",\"values\":{\"items\":[{\"value\":\"on-demand\"}]}},{\"key\":\"aws:Resource
\",\"values\":{\"items\":[{\"value\":\"instance/*\"}]}},{\"key\":\"aws:Account
```

```
\",\"values\":{\"items\":[{\"value\":\"111122223333\"}]}},{\"key\":
\"ec2:AvailabilityZone\",\"values\":{\"items\":[{\"value\":\"us-east-1f\"}]}},
{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items\":[{\"value\":\"false\"}]}},
{\"key\":\"ec2:IsLaunchTemplateResource\",\"values\":{\"items\":[{\"value\":
\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":{\"items\":[{\"value
\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values\":{\"items\":
[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-1\"}]}},{\"key\":\"ec2:MetadataHttpEndpoint\",\"values\":{\"items
\":[{\"value\":\"enabled\"}]}},{\"key\":\"aws:Service\",\"values\":{\"items
\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items\":
[{\"value\":\"*\"}]}},{\"key\":\"ec2:MetadataHttpTokens\",\"values\":{\"items
\":[{\"value\":\"required\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items
\":[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items
\":[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items
\":[{\"value\":\"us-east-1\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":
[{\"value\":\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}]}}"
}
```

Per ulteriori informazioni, consulta la logica di valutazione delle politiche nella Guida per l'utente AWS IAM.

- Per i dettagli sull'API, consulta DecodeAuthorizationMessage in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: decodifica le informazioni aggiuntive contenute nel contenuto del messaggio codificato fornito e restituito in risposta a una richiesta. Le informazioni aggiuntive sono codificate perché i dettagli dello stato di autorizzazione possono costituire informazioni privilegiate che l'utente che ha richiesto l'azione non dovrebbe vedere.

```
Convert-STSAuthorizationMessage -EncodedMessage "...encoded message..."
```

- Per i dettagli sull'API, vedere DecodeAuthorizationMessage in AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetFederationToken`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetFederationToken`.

CLI

AWS CLI

Per restituire un set di credenziali di sicurezza temporanee utilizzando le credenziali della chiave di accesso utente IAM

L'`get-federation-token`esempio seguente restituisce un set di credenziali di sicurezza temporanee (costituite da un ID di chiave di accesso, una chiave di accesso segreta e un token di sicurezza) per un utente. È necessario chiamare l'`GetFederationToken`operazione utilizzando le credenziali di sicurezza a lungo termine di un utente IAM.

```
aws sts get-federation-token \
    --name Bob \
    --policy file://myfile.json \
    --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
    --duration-seconds 900
```

Contenuto di `myfile.json`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:Describe*",
            "Resource": "*"
        }
    ]
}
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "EXAMPLEpZ2luX2VjEGoaCXVzLXdlc3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QQunWMTfKq0DCOP/////////
wEQAxoMNDUyOTI1MTcwNTA3Igxuy3AOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
        "Expiration": "2023-12-20T02:06:07+00:00"
    },
    "FederatedUser": {
        "FederatedUserId": "111122223333:Bob",
        "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
    },
    "PackedPolicySize": 36
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetFederationToken in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: richiede un token federato valido per un'ora utilizzando «Bob» come nome dell'utente federato. Questo nome può essere usato per fare riferimento al nome utente

federato in una policy basata sulle risorse (ad esempio una bucket policy di Amazon S3). La policy IAM fornita, in formato JSON, viene utilizzata per definire le autorizzazioni disponibili per l'utente IAM. La policy fornita non può concedere più autorizzazioni di quelle concesse all'utente richiedente, e le autorizzazioni finali per l'utente federato sono il set più restrittivo in base all'intersezione tra la policy passata e la policy utente IAM.

```
Get-STSFederationToken -Name "Bob" -Policy "...JSON policy..." -DurationInSeconds
 3600
```

- Per i dettagli sull'API, vedere Token in Cmdlet Reference. GetFederation AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetSessionToken`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetSessionToken`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Recupero di un token di sessione che richiede un token MFA

CLI

AWS CLI

Come ottenere un set di credenziali a breve termine per un'identità IAM

il comando `get-session-token` seguente recupera un set di credenziali a breve termine per l'identità IAM che esegue la chiamata. Le credenziali risultanti possono essere utilizzate per richieste in cui l'autenticazione a più fattori (MFA) è richiesta dalla policy. Le credenziali scadono 15 minuti dopo la loro generazione.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
```

```
    --token-code 123456
```

Output:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza temporanee nella AWS Guida per l'utente di IAM.

- Per i dettagli sull'API, consulta GetSessionToken in AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce un'**Amazon.RuntimeAWSCredentials**istanza contenente credenziali temporanee valide per un determinato periodo di tempo. Le credenziali utilizzate per richiedere credenziali temporanee vengono dedotte dalle impostazioni predefinite correnti della shell. Per specificare altre credenziali, utilizzate i parametri - o - /. ProfileName AccessKey SecretKey

```
Get-STSSessionToken
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 2: restituisce un'**Amazon.RuntimeAWSCredentials**istanza contenente credenziali temporanee valide per un'ora. Le credenziali utilizzate per effettuare la richiesta vengono ottenute dal profilo specificato.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

Esempio 3: restituisce un'**Amazon.RuntimeAWSCredentials**istanza contenente credenziali temporanee valide per un'ora utilizzando il numero di identificazione del dispositivo MFA associato all'account le cui credenziali sono specificate nel profilo 'myprofilename' e il valore fornito dal dispositivo.

```
Get-STSSessionToken -DurationInSeconds 3600 -ProfileName myprofile -SerialNumber
 YourMFADeviceSerialNumber -TokenCode 123456
```

Output:

```
AccessKeyId                              Expiration
 SecretAccessKey                         SessionToken
-----------                              ----------
 ---------------                         ------------
EXAMPLEACCESSKEYID                       2/16/2015 9:12:28 PM
 examplesecretaccesskey...               SamPleTokeN.....
```

- Per i dettagli sull'API, vedere Token in Cmdlet Reference. GetSession AWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un token di sessione passando un token MFA e utilizzalo per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
```

```
    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

- Per i dettagli sull'API, consulta GetSessionToken in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l' AWS STS utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni AWS STS con gli AWS SDK. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno. AWS STS Ogni scenario include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Assumi un ruolo IAM che richiede un token MFA con l' AWS STS utilizzo di un SDK AWS
- Costruisci un URL con AWS STS per utenti federati utilizzando un SDK AWS
- Ottieni un token di sessione che richiede un token MFA AWS STS utilizzando un SDK AWS

Assumi un ruolo IAM che richiede un token MFA con l' AWS STS utilizzo di un SDK AWS

L'esempio di codice seguente mostra come assumere un ruolo che richiede un token MFA.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Creare un ruolo IAM che conceda l'autorizzazione per elencare i bucket Amazon S3.
- Creare un utente IAM che abbia il permesso di assumere il ruolo solo quando vengono fornite le credenziali MFA.
- Registrare un dispositivo MFA per l'utente.
- Assumere il ruolo ed elencare i bucket Amazon S3 utilizzando le credenziali temporanee.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare un utente IAM, registrare un dispositivo MFA e creare un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3. L'utente dispone dei diritti soltanto per assumere il ruolo.

```
def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates a new virtual MFA device.
    Displays the QR code to seed the device.
    Asks for two codes from the MFA device.
    Registers the MFA device for the user.
    Creates an access key pair for the user.
    Creates a role with a policy that lets the user assume the role and requires
 MFA.
    Creates a policy that allows listing Amazon S3 buckets.
    Attaches the policy to the role.
    Creates an inline policy for the user that lets the user assume the role.

    For demonstration purposes, the user is created in the same account as the
 role,
    but in practice the user would likely be from another account.

    Any MFA device that can scan a QR code will work with this demonstration.
    Common choices are mobile apps like LastPass Authenticator,
```

```
    Microsoft Authenticator, or Google Authenticator.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
resource
                         that has permissions to create users, roles, and
policies
                         in the account.
    :return: The newly created user, user key, virtual MFA device, and role.
    """
    user = iam_resource.create_user(UserName=unique_name("user"))
    print(f"Created user {user.name}.")

    virtual_mfa_device = iam_resource.create_virtual_mfa_device(
        VirtualMFADeviceName=unique_name("mfa")
    )
    print(f"Created virtual MFA device {virtual_mfa_device.serial_number}")

    print(
        f"Showing the QR code for the device. Scan this in the MFA app of your "
        f"choice."
    )
    with open("qr.png", "wb") as qr_file:
        qr_file.write(virtual_mfa_device.qr_code_png)
    webbrowser.open(qr_file.name)

    print(f"Enter two consecutive code from your MFA device.")
    mfa_code_1 = input("Enter the first code: ")
    mfa_code_2 = input("Enter the second code: ")
    user.enable_mfa(
        SerialNumber=virtual_mfa_device.serial_number,
        AuthenticationCode1=mfa_code_1,
        AuthenticationCode2=mfa_code_2,
    )
    os.remove(qr_file.name)
    print(f"MFA device is registered with the user.")

    user_key = user.create_access_key_pair()
    print(f"Created access key pair for user.")

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    role = iam_resource.create_role(
        RoleName=unique_name("role"),
```

```
        AssumeRolePolicyDocument=json.dumps(
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": user.arn},
                        "Action": "sts:AssumeRole",
                        "Condition": {"Bool": {"aws:MultiFactorAuthPresent":
True}},
                    }
                ],
            }
        ),
    )
    print(f"Created role {role.name} that requires MFA.")

    policy = iam_resource.create_policy(
        PolicyName=unique_name("policy"),
        PolicyDocument=json.dumps(
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "s3:ListAllMyBuckets",
                        "Resource": "arn:aws:s3:::*",
                    }
                ],
            }
        ),
    )
    role.attach_policy(PolicyArn=policy.arn)
    print(f"Created policy {policy.policy_name} and attached it to the role.")

    user.create_policy(
        PolicyName=unique_name("user-policy"),
        PolicyDocument=json.dumps(
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "sts:AssumeRole",
```

```
                        "Resource": role.arn,
                    }
                ],
            }
        ),
    )
    print(
        f"Created an inline policy for {user.name} that lets the user assume "
        f"the role."
    )

    print("Give AWS time to propagate these new resources and connections.",
end="")
    progress_bar(10)

    return user, user_key, virtual_mfa_device, role
```

Dimostra che non è consentito assumere il ruolo senza un token MFA.

```
def try_to_assume_role_without_mfa(assume_role_arn, session_name, sts_client):
    """
    Shows that attempting to assume the role without sending MFA credentials
results
    in an AccessDenied error.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role to assume.
    :param session_name: The name of the STS session.
    :param sts_client: A Boto3 STS instance that has permission to assume the
role.
    """
    print(f"Trying to assume the role without sending MFA credentials...")
    try:
        sts_client.assume_role(RoleArn=assume_role_arn,
RoleSessionName=session_name)
        raise RuntimeError("Expected AccessDenied error.")
    except ClientError as error:
        if error.response["Error"]["Code"] == "AccessDenied":
            print("Got AccessDenied.")
        else:
            raise
```

Assumere il ruolo che concede l'autorizzazione per elencare i bucket Amazon S3, passando il token MFA richiesto e mostrare che i bucket possono essere elencati.

```
def list_buckets_from_assumed_role_with_mfa(
    assume_role_arn, session_name, mfa_serial_number, mfa_totp, sts_client
):
    """
    Assumes a role from another account and uses the temporary credentials from
    that role to list the Amazon S3 buckets that are owned by the other account.
    Requires an MFA device serial number and token.

    The assumed role must grant permission to list the buckets in the other
 account.

    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an ARN.
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    response = sts_client.assume_role(
        RoleArn=assume_role_arn,
        RoleSessionName=session_name,
        SerialNumber=mfa_serial_number,
        TokenCode=mfa_totp,
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    s3_resource = boto3.resource(
        "s3",
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )
```

```
    print(f"Listing buckets for the assumed role's account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

Elimina le risorse create per la demo.

```
def teardown(user, virtual_mfa_device, role):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo role.
    """
    for attached in role.attached_policies.all():
        policy_name = attached.policy_name
        role.detach_policy(PolicyArn=attached.arn)
        attached.delete()
        print(f"Detached and deleted {policy_name}.")
    role.delete()
    print(f"Deleted {role.name}.")
    for user_pol in user.policies.all():
        user_pol.delete()
        print("Deleted inline user policy.")
    for key in user.access_keys.all():
        key.delete()
        print("Deleted user's access key.")
    for mfa in user.mfa_devices.all():
        mfa.disassociate()
    virtual_mfa_device.delete()
    user.delete()
    print(f"Deleted {user.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
```

```
    print("-" * 88)
    print(
        f"Welcome to the AWS Security Token Service assume role demo, "
        f"starring multi-factor authentication (MFA)!"
    )
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user, user_key, virtual_mfa_device, role = setup(iam_resource)
    print(f"Created {user.name} and {role.name}.")
    try:
        sts_client = boto3.client(
            "sts", aws_access_key_id=user_key.id,
aws_secret_access_key=user_key.secret
        )
        try_to_assume_role_without_mfa(role.arn, "demo-sts-session", sts_client)
        mfa_totp = input("Enter the code from your registered MFA device: ")
        list_buckets_from_assumed_role_with_mfa(
            role.arn,
            "demo-sts-session",
            virtual_mfa_device.serial_number,
            mfa_totp,
            sts_client,
        )
    finally:
        teardown(user, virtual_mfa_device, role)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Costruisci un URL con AWS STS per utenti federati utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Creare un IAM ruolo che conceda l'accesso in sola lettura alle risorse Amazon S3 dell'account corrente.

- Ottieni un token di sicurezza dall'endpoint della AWS federazione.
- Creare un URL che possa essere utilizzato per accedere alla console con credenziali federate.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un ruolo che conceda l'accesso in sola lettura alle risorse Amazon S3 dell'account corrente.

```
def setup(iam_resource):
    """
    Creates a role that can be assumed by the current user.
    Attaches a policy that allows only Amazon S3 read-only access.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
 instance
                         that has the permission to create a role.
    :return: The newly created role.
    """
    role = iam_resource.create_role(
        RoleName=unique_name("role"),
        AssumeRolePolicyDocument=json.dumps(
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": iam_resource.CurrentUser().arn},
                        "Action": "sts:AssumeRole",
                    }
                ],
            }
        ),
    )
```

```
    role.attach_policy(PolicyArn="arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess")
    print(f"Created role {role.name}.")

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return role
```

Ottieni un token di sicurezza dall'endpoint AWS federativo e crea un URL che può essere utilizzato per accedere alla console con credenziali federate.

```
def construct_federated_url(assume_role_arn, session_name, issuer, sts_client):
    """
    Constructs a URL that gives federated users direct access to the AWS
 Management
    Console.

    1. Acquires temporary credentials from AWS Security Token Service (AWS STS)
 that
       can be used to assume a role with limited permissions.
    2. Uses the temporary credentials to request a sign-in token from the
       AWS federation endpoint.
    3. Builds a URL that can be used in a browser to navigate to the AWS
 federation
       endpoint, includes the sign-in token for authentication, and redirects to
       the AWS Management Console with permissions defined by the role that was
       specified in step 1.

    :param assume_role_arn: The role that specifies the permissions that are
 granted.
                            The current user must have permission to assume the
 role.
    :param session_name: The name for the STS session.
    :param issuer: The organization that issues the URL.
    :param sts_client: A Boto3 STS instance that can assume the role.
    :return: The federated URL.
    """
    response = sts_client.assume_role(
```

```
        RoleArn=assume_role_arn, RoleSessionName=session_name
    )
    temp_credentials = response["Credentials"]
    print(f"Assumed role {assume_role_arn} and got temporary credentials.")

    session_data = {
        "sessionId": temp_credentials["AccessKeyId"],
        "sessionKey": temp_credentials["SecretAccessKey"],
        "sessionToken": temp_credentials["SessionToken"],
    }
    aws_federated_signin_endpoint = "https://signin.aws.amazon.com/federation"

    # Make a request to the AWS federation endpoint to get a sign-in token.
    # The requests.get function URL-encodes the parameters and builds the query
string
    # before making the request.
    response = requests.get(
        aws_federated_signin_endpoint,
        params={
            "Action": "getSigninToken",
            "SessionDuration": str(datetime.timedelta(hours=12).seconds),
            "Session": json.dumps(session_data),
        },
    )
    signin_token = json.loads(response.text)
    print(f"Got a sign-in token from the AWS sign-in federation endpoint.")

    # Make a federated URL that can be used to sign into the AWS Management
Console.
    query_string = urllib.parse.urlencode(
        {
            "Action": "login",
            "Issuer": issuer,
            "Destination": "https://console.aws.amazon.com/",
            "SigninToken": signin_token["SigninToken"],
        }
    )
    federated_url = f"{aws_federated_signin_endpoint}?{query_string}"
    return federated_url
```

Elimina le risorse create per la demo.

```
def teardown(role):
    """
    Removes all resources created during setup.

    :param role: The demo role.
    """
    for attached in role.attached_policies.all():
        role.detach_policy(PolicyArn=attached.arn)
        print(f"Detached {attached.policy_name}.")
    role.delete()
    print(f"Deleted {role.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(f"Welcome to the AWS Security Token Service federated URL demo.")
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    role = setup(iam_resource)
    sts_client = boto3.client("sts")
    try:
        federated_url = construct_federated_url(
            role.arn, "AssumeRoleDemoSession", "example.org", sts_client
        )
        print(
            "Constructed a federated URL that can be used to connect to the "
            "AWS Management Console with role-defined permissions:"
        )
        print("-" * 88)
        print(federated_url)
        print("-" * 88)
        _ = input(
            "Copy and paste the above URL into a browser to open the AWS "
            "Management Console with limited permissions. When done, press "
            "Enter to clean up and complete this demo."
        )
```

```
    finally:
        teardown(role)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta AssumeRole AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ottieni un token di sessione che richiede un token MFA AWS STS utilizzando un SDK AWS

L'esempio di codice seguente mostra come ottenere un token di sessione che richiede un token MFA.

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

- Creare un ruolo IAM che conceda l'autorizzazione per elencare i bucket Amazon S3.
- Creare un utente IAM che abbia il permesso di assumere il ruolo solo quando vengono fornite le credenziali MFA.
- Registrare un dispositivo MFA per l'utente.
- Fornire le credenziali MFA per ottenere un token di sessione e utilizzare le credenziali temporanee per elencare i bucket S3.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare un utente IAM, registrare un dispositivo MFA e creare un ruolo che conceda l'autorizzazione per consentire all'utente di elencare i bucket Amazon S3 solo quando si utilizzano le credenziali MFA.

```
def setup(iam_resource):
    """
    Creates a new user with no permissions.
    Creates a new virtual multi-factor authentication (MFA) device.
    Displays the QR code to seed the device.
    Asks for two codes from the MFA device.
    Registers the MFA device for the user.
    Creates an access key pair for the user.
    Creates an inline policy for the user that lets the user list Amazon S3
 buckets,
    but only when MFA credentials are used.

    Any MFA device that can scan a QR code will work with this demonstration.
    Common choices are mobile apps like LastPass Authenticator,
    Microsoft Authenticator, or Google Authenticator.

    :param iam_resource: A Boto3 AWS Identity and Access Management (IAM)
 resource
                         that has permissions to create users, MFA devices, and
                         policies in the account.
    :return: The newly created user, user key, and virtual MFA device.
    """
    user = iam_resource.create_user(UserName=unique_name("user"))
    print(f"Created user {user.name}.")

    virtual_mfa_device = iam_resource.create_virtual_mfa_device(
        VirtualMFADeviceName=unique_name("mfa")
    )
```

```
    print(f"Created virtual MFA device {virtual_mfa_device.serial_number}")

    print(
        f"Showing the QR code for the device. Scan this in the MFA app of your "
        f"choice."
    )
    with open("qr.png", "wb") as qr_file:
        qr_file.write(virtual_mfa_device.qr_code_png)
    webbrowser.open(qr_file.name)

    print(f"Enter two consecutive code from your MFA device.")
    mfa_code_1 = input("Enter the first code: ")
    mfa_code_2 = input("Enter the second code: ")
    user.enable_mfa(
        SerialNumber=virtual_mfa_device.serial_number,
        AuthenticationCode1=mfa_code_1,
        AuthenticationCode2=mfa_code_2,
    )
    os.remove(qr_file.name)
    print(f"MFA device is registered with the user.")

    user_key = user.create_access_key_pair()
    print(f"Created access key pair for user.")

    print(f"Wait for user to be ready.", end="")
    progress_bar(10)

    user.create_policy(
        PolicyName=unique_name("user-policy"),
        PolicyDocument=json.dumps(
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": "s3:ListAllMyBuckets",
                        "Resource": "arn:aws:s3:::*",
                        "Condition": {"Bool": {"aws:MultiFactorAuthPresent":
True}},
                    }
                ],
            }
        ),
    )
```

```
    print(
        f"Created an inline policy for {user.name} that lets the user list
 buckets, "
        f"but only when MFA credentials are present."
    )

    print("Give AWS time to propagate these new resources and connections.",
 end="")
    progress_bar(10)

    return user, user_key, virtual_mfa_device
```

Recuperare le credenziali di sessione temporanee passando un token MFA e utilizzarle per elencare i bucket Amazon S3 per l'account.

```
def list_buckets_with_session_token_with_mfa(mfa_serial_number, mfa_totp,
 sts_client):
    """
    Gets a session token with MFA credentials and uses the temporary session
    credentials to list Amazon S3 buckets.

    Requires an MFA device serial number and token.

    :param mfa_serial_number: The serial number of the MFA device. For a virtual
 MFA
                              device, this is an Amazon Resource Name (ARN).
    :param mfa_totp: A time-based, one-time password issued by the MFA device.
    :param sts_client: A Boto3 STS instance that has permission to assume the
 role.
    """
    if mfa_serial_number is not None:
        response = sts_client.get_session_token(
            SerialNumber=mfa_serial_number, TokenCode=mfa_totp
        )
    else:
        response = sts_client.get_session_token()
    temp_credentials = response["Credentials"]

    s3_resource = boto3.resource(
        "s3",
```

```
        aws_access_key_id=temp_credentials["AccessKeyId"],
        aws_secret_access_key=temp_credentials["SecretAccessKey"],
        aws_session_token=temp_credentials["SessionToken"],
    )

    print(f"Buckets for the account:")
    for bucket in s3_resource.buckets.all():
        print(bucket.name)
```

Elimina le risorse create per la demo.

```
def teardown(user, virtual_mfa_device):
    """
    Removes all resources created during setup.

    :param user: The demo user.
    :param role: The demo MFA device.
    """
    for user_pol in user.policies.all():
        user_pol.delete()
        print("Deleted inline user policy.")
    for key in user.access_keys.all():
        key.delete()
        print("Deleted user's access key.")
    for mfa in user.mfa_devices.all():
        mfa.disassociate()
    virtual_mfa_device.delete()
    user.delete()
    print(f"Deleted {user.name}.")
```

Esegui questo scenario utilizzando le funzioni definite in precedenza.

```
def usage_demo():
    """Drives the demonstration."""
    print("-" * 88)
    print(
        f"Welcome to the AWS Security Token Service assume role demo, "
```

```
        f"starring multi-factor authentication (MFA)!"
    )
    print("-" * 88)
    iam_resource = boto3.resource("iam")
    user, user_key, virtual_mfa_device = setup(iam_resource)
    try:
        sts_client = boto3.client(
            "sts", aws_access_key_id=user_key.id,
aws_secret_access_key=user_key.secret
        )
        try:
            print("Listing buckets without specifying MFA credentials.")
            list_buckets_with_session_token_with_mfa(None, None, sts_client)
        except ClientError as error:
            if error.response["Error"]["Code"] == "AccessDenied":
                print("Got expected AccessDenied error.")
        mfa_totp = input("Enter the code from your registered MFA device: ")
        list_buckets_with_session_token_with_mfa(
            virtual_mfa_device.serial_number, mfa_totp, sts_client
        )
    finally:
        teardown(user, virtual_mfa_device)
        print("Thanks for watching!")
```

- Per i dettagli sull'API, consulta GetSessionToken in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di IAM con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in IAM e AWS STS

La sicurezza del cloud ha AWS la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili a AWS Identity and Access Management (IAM), consulta AWS Services in Scope by Compliance Program AWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizzano AWS Identity and Access Management (IAM) e AWS Security Token Service (AWS STS). I seguenti argomenti mostrano come configurare IAM e AWS STS soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse IAM.

Indice

AWS credenziali di sicurezza

Quando interagisci con AWS, specifichi le tue credenziali di AWS sicurezza per verificare chi sei e se disponi dell'autorizzazione ad accedere alle risorse che stai richiedendo. AWS utilizza le credenziali di sicurezza per autenticare e autorizzare le richieste.

Ad esempio, se si desidera scaricare un file protetto da un bucket Amazon Simple Storage Service (Amazon S3), è necessario che le credenziali consentano tale accesso. Se le tue credenziali non mostrano che sei autorizzato a scaricare il file, AWS respinge la tua richiesta. Tuttavia, le tue credenziali di AWS sicurezza non sono necessarie per scaricare un file in un bucket Amazon S3 condiviso pubblicamente.

Esistono diversi tipi di utenti in. AWS Tutti AWS gli utenti dispongono di credenziali di sicurezza. Sono presenti il proprietario dell'account (utente root) utenti in Centro identità AWS IAM, gli utenti federati e gli utenti IAM.

Gli utenti dispongono di credenziali di sicurezza a lungo termine o temporanee. L'utente root, l'utente IAM e le chiavi di accesso dispongono di credenziali di sicurezza a lungo termine che non scadono. Per proteggere le credenziali a lungo termine, è consigliabile disporre di procedure per gestire le chiavi di accesso, modificare le password e abilitare l'MFA.

I ruoli IAM e utenti in Centro identità AWS IAM gli utenti federati dispongono di credenziali di sicurezza temporanee. Le credenziali di sicurezza temporanee scadono dopo un periodo di tempo definito o quando l'utente termina la sessione. Le credenziali temporanee funzionano quasi esattamente come le credenziali a lungo termine, con le seguenti differenze:

- Le credenziali di sicurezza provvisorie sono a breve termine, come implica il nome. Possono essere configurate per durare ovunque per pochi minuti o diverse ore. Una volta scadute, le credenziali AWS non le riconosce più né consente alcun tipo di accesso alle richieste API effettuate con esse.
- Le credenziali di sicurezza temporanee non sono archiviate con l'utente, ma vengono generate dinamicamente e fornite all'utente quando richiesto. Quando (o anche prima) le credenziali di sicurezza temporanee scadono, l'utente può richiedere nuove credenziali, purché l'utente che le richiede abbia ancora le autorizzazioni per farlo.

Di conseguenza, le credenziali temporanee presentano i seguenti vantaggi rispetto alle credenziali a lungo termine:

- Non è necessario distribuire o incorporare credenziali di AWS sicurezza a lungo termine in un'applicazione.
- È possibile fornire l'accesso alle AWS risorse agli utenti senza dover definire un' AWS identità per loro. Le credenziali provvisorie sono la base dei ruoli e della federazione delle identità.
- Le credenziali di sicurezza temporanee hanno una durata limitata, perciò non è necessario aggiornarle o revocarle in modo esplicito quando non sono più necessarie. Dopo che le credenziali di sicurezza temporanee scadono, non possono essere riutilizzate. È possibile specificare quando scadono le credenziali, fino a un limite massimo.

Considerazioni relative alla sicurezza

Ti consigliamo di prendere in considerazione le informazioni seguenti nel momento in cui stabilisci le disposizioni di sicurezza per il tuo Account AWS:

- Quando crei un account Account AWS, creiamo l'utente root dell'account. Le credenziali dell'utente root (proprietario dell'account) consentono il pieno accesso a tutte le risorse nell'account. La prima operazione da eseguire con l'utente root è concedere a un altro utente le autorizzazioni amministrative Account AWS in modo da ridurre al minimo l'utilizzo dell'utente root.
- Non puoi utilizzare policy IAM per negare esplicitamente all'utente root l'accesso alle risorse. È possibile utilizzare solo una policy AWS Organizations di controllo del servizio (SCP) per limitare le autorizzazioni dell'utente root.
- Se dimentichi o perdi la password dell'utente root, dovrai accedere all'indirizzo e-mail associato al tuo account per reimpostarla.
- Se perdi le chiavi di accesso dell'utente root, devi essere in grado di accedere al tuo account come utente root per crearne di nuove.
- Non utilizzare l'utente root per le attività quotidiane. Utilizzalo per eseguire le attività che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono il tuo accesso come utente root, consulta la pagina Attività che richiedono credenziali dell'utente root.
- Le credenziali di sicurezza sono specifiche dell'account. Se hai accesso a più Account AWS, avrai credenziali separate per ogni account.
- Le politiche determinano le azioni che un utente, un ruolo o un membro di un gruppo di utenti può eseguire, su quali AWS risorse e in quali condizioni. Utilizzando le policy è possibile controllare in modo sicuro l'accesso Servizi AWS e le risorse presenti in. Account AWS Se devi modificare o revocare le autorizzazioni in risposta a un evento di sicurezza, puoi eliminare o modificare le policy anziché modificare direttamente l'identità.

- Assicurati di salvare in un luogo sicuro le credenziali di accesso per il tuo utente IAM per l'accesso di emergenza e tutte le chiavi di accesso che hai creato per l'accesso programmatico. Se perdi le chiavi di accesso, dovrai accedere al tuo account e crearne di nuove.
- Ti consigliamo vivamente di utilizzare le credenziali temporanee fornite dai ruoli IAM e dagli utenti federati anziché quelle a lungo termine fornite dagli utenti IAM e dalle chiavi di accesso.

Identità federata

Le identità federate sono utenti con identità esterne a cui vengono concesse AWS credenziali temporanee che possono utilizzare per accedere a risorse sicure. Account AWS Le identità esterne possono provenire da un archivio di identità aziendali (ad esempio LDAP o Windows Active Directory) o da terze parti (ad esempio Login with Amazon, Facebook e Google). Le identità federate non accedono né accedono al portale. AWS Management Console AWS

Per consentire alle identità federate di accedere ad AWS, devi creare un URL personalizzato che includa `https://signin.aws.amazon.com/federation`. Per ulteriori informazioni, consulta Abilitazione dell'accesso personalizzato da parte di un broker di identità alla AWS console.

Per ulteriori informazioni sulle identità federate, consulta la pagina Provider di identità e federazione.

Autenticazione a più fattori (MFA)

L'autenticazione a più fattori (MFA) offre un ulteriore livello di sicurezza per gli utenti che possono accedere al tuo Account AWS. Per una maggiore sicurezza, consigliamo di richiedere l'MFA sulle credenziali di Utente root dell'account AWS e di tutti gli utenti IAM. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

Quando attivi la MFA e accedi al tuo Account AWS, ti vengono richieste le credenziali di accesso, oltre a una risposta generata da un dispositivo MFA, ad esempio un codice, un tocco o un tocco o una scansione biometrica. Quando aggiungi la tecnologia MFA, Account AWS le impostazioni e le risorse sono più sicure.

Per impostazione predefinita, l'MFA non è attivata. Puoi attivare e gestire i dispositivi MFA per Utente root dell'account AWS visitando la pagina Credenziali di sicurezza o il pannello di controllo IAM nella AWS Management Console. Per ulteriori informazioni sull'attivazione dell'MFA per gli utenti IAM, consulta la pagina Abilitazione dei dispositivi MFA per gli utenti in AWS.

Per ulteriori informazioni sull'accesso con dispositivi di autenticazione a più fattori (MFA), consulta la pagina Utilizzo di dispositivi MFA con la pagina di accesso IAM.

Accesso programmatico

Fornisci le tue chiavi di AWS accesso per effettuare chiamate programmatiche AWS o per utilizzare la AWS Command Line Interface sala operatoria. AWS Tools for PowerShell Consigliamo di utilizzare chiavi di accesso a breve termine quando possibile.

Quando crei una chiavi di accesso a lungo termine, crei anche l'ID chiave di accesso (ad esempio, `AKIAIOSFODNN7EXAMPLE`) e la chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). La chiave di accesso segreta può essere scaricata solo nel momento in cui viene creata. Se non si scarica la chiave di accesso segreta o se viene smarrita, è necessario crearne una nuova.

In molti scenari, non è necessaria una chiave di accesso a lungo termine a validità illimitata (come accade invece quando si creano le chiavi di accesso per un utente IAM). Al contrario, è possibile creare ruoli IAM e generare credenziali di sicurezza temporanee. Tali credenziali di sicurezza temporanee includono un ID chiave di accesso e una chiave di accesso segreta, ma includono anche un token di sicurezza che ne indica la scadenza. Dopo che scadono, non sono più valide.

Gli ID delle chiavi di accesso che iniziano con `AKIA` sono chiavi di accesso a lungo termine per un utente IAM o un utente Account AWS root. Gli ID delle chiavi di accesso che iniziano con `ASIA` sono credenziali temporanee, chiavi di accesso create utilizzando AWS STS le operazioni.

Gli utenti necessitano dell'accesso programmatico se desiderano interagire con l' AWS esterno di. AWS Management Console Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporane e per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per

Quale utente necessita dell'accesso programmatico?	Per	Come
		l'utente.AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento agli AWS SDK e agli strumenti.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Segui le istruzioni in Uso delle credenziali temporanee con AWS risorse nella Guida per l'utente IAM.
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli AWS SDK e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimen to agli SDK e agli AWS strumenti. • Per le AWS API, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Alternative alle chiavi di accesso a lungo termine

Per numerosi casi d'uso comuni, esistono delle alternative alle chiavi di accesso a lungo termine. Per migliorare la sicurezza del tuo account, considera quanto segue.

- Non incorporate chiavi di accesso a lungo termine e chiavi di accesso segrete nel codice dell'applicazione o in un repository di codice: utilizzate invece una o altre soluzioni di gestione dei segreti AWS Secrets Manager, in modo da non dover codificare le chiavi in testo non crittografato. L'applicazione o il client possono quindi recuperare i segreti quando necessario. Per ulteriori informazioni, consulta Cos'è? AWS Secrets Manager nella Guida AWS Secrets Manager per l'utente.

- Quando possibile, utilizza ruoli IAM per generare credenziali di sicurezza temporanee: usa sempre dei meccanismi per emettere credenziali di sicurezza temporanee anziché chiavi di accesso a lungo termine. Le credenziali di sicurezza temporanee sono più sicure perché non sono archiviate con l'utente ma vengono generate dinamicamente e fornite all'utente quando richiesto. Le credenziali di sicurezza temporanee hanno una durata limitata, quindi non è necessario gestirle o aggiornarle. I meccanismi che forniscono chiavi di accesso temporanee includono i ruoli IAM o l'autenticazione di un utente IAM Identity Center. Per le macchine che funzionano all'esterno dell' AWS utente, è possibile utilizzare AWS Identity and Access Management Roles Anywhere.
- Utilizza alternative alle chiavi di accesso a lungo termine per AWS Command Line Interface (AWS CLI) o la **`aws-shell`**: le alternative includono quanto segue.
  - AWS CloudShellè una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsoleÈ possibile eseguire AWS CLI comandi Servizi AWS tramite la shell preferita (Bash, Powershell o Z shell). Quando esegui questa operazione, non devi scaricare o installare strumenti a riga di comando. Per ulteriori informazioni, consulta Che cos'è AWS CloudShell? nella Guida per l'utente di AWS CloudShell .
  - AWS CLI Integrazione della versione 2 con AWS IAM Identity Center (IAM Identity Center). Puoi autenticare gli utenti e fornire credenziali a breve termine per eseguire AWS CLI i comandi. Per ulteriori informazioni, consulta Integrazione AWS CLI con IAM Identity Center nella Guida per l'AWS IAM Identity Center utente e Configurazione per l'utilizzo di IAM Identity Center nella Guida AWS CLI per l'utente.AWS Command Line Interface
- Non creare chiavi di accesso a lungo termine per utenti umani che devono accedere alle applicazioni, altrimenti Servizi AWS IAM Identity Center può generare credenziali di accesso temporanee a cui possono accedere gli utenti IdP esterni. Servizi AWS In questo modo si elimina la necessità di creare e gestire credenziali a lungo termine in IAM. In IAM Identity Center, crea un

set di autorizzazioni di IAM Identity Center che conceda l'accesso agli utenti IdP esterni. Quindi assegna un gruppo da IAM Identity Center al set di autorizzazioni selezionato. Account AWS Per ulteriori informazioni, consulta What is AWS IAM Identity Center, Connect to your identity provider esterno e Set di autorizzazioni nella Guida per l'AWS IAM Identity Center utente.

- Non archiviate le chiavi di accesso a lungo termine all'interno di un servizio di AWS elaborazione, ma assegnate invece un ruolo IAM alle risorse di calcolo. Ciò fornisce automaticamente le credenziali temporanee per concedere l'accesso. Ad esempio, quando crei un profilo di istanza collegato a un'istanza Amazon EC2, puoi assegnare un AWS ruolo all'istanza e renderla disponibile per tutte le sue applicazioni. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza Amazon EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2.

Accesso AWS utilizzando le tue credenziali AWS

AWS richiede diversi tipi di credenziali di sicurezza, a seconda della modalità di accesso AWS e del tipo di AWS utente. Ad esempio, puoi usare credenziali di accesso per la AWS Management Console mentre utilizzi le chiavi di accesso per fare chiamate programmatiche ad AWS. Inoltre, ogni identità utilizzata, che si tratti dell'utente root dell'account, di un utente AWS Identity and Access Management (IAM), di un utente o di un' AWS IAM Identity Center identità federata, contiene credenziali univoche. AWS

Per step-by-step istruzioni su come accedere in AWS base al tipo di utente, consulta Come accedere nella Guida per AWS l'utente di AWS accesso.

AWS linee guida per l'audit di sicurezza

Controlla periodicamente la configurazione di sicurezza per accertarti che soddisfi i requisiti aziendali attuali. Grazie ai controlli puoi rimuovere ruoli, gruppi, policy e utenti IAM non necessari, inoltre puoi accertarti che gli utenti e il software dispongano solo delle autorizzazioni necessarie.

Di seguito sono riportate le linee guida per la revisione e il monitoraggio sistematici AWS delle risorse alla ricerca delle migliori pratiche di sicurezza.

> **Tip**
>
> Puoi monitorare l'uso di IAM in relazione alle best practice sulla sicurezza utilizzando AWS Security Hub. Security Hub utilizza controlli di sicurezza per valutare le configurazioni delle

risorse e gli standard di sicurezza per aiutarti a rispettare vari framework di conformità. Per ulteriori informazioni sull'utilizzo di Security Hub per la valutazione delle risorse IAM, consulta Controlli IAM (Identity and Access Management) di AWS nella Guida per l'utente di AWS Security Hub .

Indice

Quando è necessario eseguire un controllo di sicurezza

Controlla la configurazione di sicurezza nelle seguenti situazioni:

- Periodicamente. Come best practice per la sicurezza, segui la procedura descritta in questo documento a intervalli regolari.
- In caso di cambiamenti all'interno dell'organizzazione, ad esempio persone che lasciano l'azienda.
- Se hai smesso di utilizzare uno o più singoli AWS servizi, verifica di aver rimosso le autorizzazioni di cui gli utenti del tuo account non hanno più bisogno.
- Se hai aggiunto o rimosso software nei tuoi account, ad esempio applicazioni su istanze Amazon EC2, AWS OpsWorks stack, modelli, ecc. AWS CloudFormation
- Se sospetti che una persona non autorizzata possa aver eseguito l'accesso al tuo account.

Linee guida per l'audit

Quando verifichi la configurazione di sicurezza del tuo account, segui queste linee guida:

- Sii meticoloso. Esamina tutti gli elementi della configurazione di sicurezza, inclusi quelli che utilizzi raramente.
- Non dare nulla per scontato. Se non conosci a sufficienza alcuni elementi della tua configurazione di sicurezza (ad esempio il motivo della presenza di una policy specifica o dell'esistenza di un ruolo), analizza i requisiti aziendali per comprendere il potenziale rischio.
- Fai in modo che le cose siano semplici. Per facilitare i controlli (e la gestione), usa gruppi IAM, ruoli IAM, schemi di denominazione coerenti e policy semplici.

Controlla le credenziali del tuo account AWS

Segui questi passaggi quando controlli le credenziali del tuo AWS account:

1. Puoi rimuovere le eventuali chiavi di accesso associate a un utente root che non utilizzi. Ti consigliamo vivamente di non utilizzare le chiavi di accesso root per il lavoro quotidiano e di utilizzare invece utenti con credenziali temporanee, ad esempio. AWS utenti in Centro identità AWS IAM
2. Se per l'account sono necessarie chiavi di accesso, occorre aggiornarle all'occorrenza.

Verifica degli utenti IAM

Eseguire questa procedura quando si verificano gli utenti IAM esistenti:

1. Elenca gli utenti, quindi elimina gli utenti che non sono necessari.
2. Rimuovi gli utenti dai gruppi a cui non richiedono l'accesso.
3. Verifica le policy associate ai gruppi in cui si trova l'utente. Per informazioni, consulta Suggerimenti per la verifica delle policy IAM.
4. Elimina le credenziali di sicurezza di cui l'utente non ha bisogno o che potrebbero essere state esposte. Ad esempio, un utente IAM utilizzato per un'applicazione non necessita di una password (necessaria solo per accedere ai AWS siti Web). Analogamente, se un utente non utilizza chiavi di accesso, non c'è motivo per cui debba averne. Per ulteriori informazioni, consulta le pagine Gestione delle password per gli utenti IAM e Gestione delle chiavi di accesso per gli utenti IAM.

   È possibile generare e scaricare un report delle credenziali che riporta tutti gli utenti IAM presenti nell'account e lo stato delle loro diverse credenziali, tra cui password, chiavi di accesso e dispositivi MFA. Per le password e le chiavi di accesso, il report sulle credenziali mostra la data e l'orario dell'ultimo utilizzo della password o della chiave di accesso. Valuta la possibilità di

rimuovere dal tuo account le credenziali che non sono state utilizzate di recente. (Non rimuovere l'utente designato per l'accesso di emergenza.) Per ulteriori informazioni, consulta Ottenere rapporti sulle credenziali per il tuo AWS account.

5. Aggiorna le password e le chiavi di accesso quando necessario per i casi d'uso che richiedono credenziali a lungo termine. Per ulteriori informazioni, consulta le pagine Gestione delle password per gli utenti IAM e Gestione delle chiavi di accesso per gli utenti IAM.
6. Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee. Se possibile, passa dagli utenti IAM agli utenti federati, ad esempio utenti in IAM Identity Center. Mantieni il numero minimo di utenti IAM necessari alle tue applicazioni.

Verifica dei gruppi IAM

Eseguire questa procedura quando si verificano i gruppi &IAM:

1. Elenca i gruppi, quindi elimina i gruppi inutilizzati.
2. Verifica gli utenti di ciascun gruppo e rimuovi gli utenti che non appartengono al gruppo in esame.
3. Verifica le policy associate al gruppo. Per informazioni, consulta Suggerimenti per la verifica delle policy IAM.

Verifica dei ruoli IAM

Quando vengono verificati i ruoli IAM, completare le seguenti operazioni:

1. Elenca i ruoli, quindi elimina i ruoli inutilizzati.
2. Esamina la policy di attendibilità del ruolo. Accertati di sapere chi ricopre il ruolo di principal e di comprendere perché tale account o utente debba essere in grado di assumere tale ruolo.
3. Consulta la policy di accesso del ruolo per essere sicuro che conceda le autorizzazioni idonee a chiunque assumi tale ruolo. Consulta Suggerimenti per la verifica delle policy IAM.

Verifica dei provider IAM per SAML e OpenID Connect (OIDC)

Se hai creato un'entità IAM per stabilire l'attendibilità con un gestore dell'identità digitale (IdP) SAML oppure OIDC, attieniti alla seguente procedura:

1. Elimina i provider inutilizzati.

2. Scarica e rivedi i documenti AWS sui metadati per ogni IdP SAML e assicurati che i documenti riflettano le tue attuali esigenze aziendali.
3. Ottieni i documenti di metadati più recenti da SAML IdPs e aggiorna il provider in IAM.

Verifica le app per dispositivi mobili

Se hai creato un'app mobile che effettua richieste a AWS, procedi nel seguente modo:

1. Accertati che l'app mobile non contenga chiavi di accesso incorporate, anche se sono archiviate crittografate.
2. Ottieni credenziali provvisorie per l'app utilizzando le API concepite a tale scopo.

> **Note**
>
> È consigliabile utilizzare Amazon Cognito per la gestione delle identità utente nell'applicazione. Questo servizio consente di autenticare gli utenti utilizzando Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Per ulteriori informazioni, consulta pool di identità in Amazon Cognito nella Guida per gli sviluppatori di Amazon Cognito.

Suggerimenti per la verifica delle policy IAM

Le policy sono potenti e ingegnose, per cui è importante studiare e comprendere le autorizzazioni concesse da ogni policy. Quando esamini le policy, utilizza le seguenti linee guida:

- Collega le policy a gruppi o ruoli anziché ai singoli utenti. Se un singolo utente dispone di una policy, assicurati di comprendere perché l'utente necessita di tale policy.
- Accertati che utenti, gruppi e ruoli IAM dispongano delle autorizzazioni necessarie e che non dispongano di autorizzazioni aggiuntive.
- Utilizza il simulatore di policy IAM per testare le policy collegate a utenti o gruppi.
- Ricorda che le autorizzazioni di un utente sono il risultato di tutte le politiche applicabili, sia politiche basate sull'identità (su utenti, gruppi o ruoli) che politiche basate sulle risorse (su risorse come bucket Amazon S3, code Amazon SQS, argomenti e chiavi di Amazon SNS). AWS KMS È importante esaminare tutte le policy che si applicano a un utente e comprendere il relativo set completo di autorizzazioni concesso.

- È importante sapere che consentire a un utente di creare un utente, un gruppo, un ruolo o una policy IAM e collegare una policy all'entità principal significa effettivamente concedere a tale utente le autorizzazioni per tutte le risorse presenti nell'account. Gli utenti che possono creare policy e collegarle a un utente, gruppo o ruolo possono assegnare a se stessi qualunque autorizzazione. In generale, non concedere a utenti o ruoli che non ritieni attendibili autorizzazioni IAM per l'accesso completo alle risorse del tuo account. Quando esegui i controlli di sicurezza, accertati che le seguenti autorizzazioni IAM siano concesse a identità attendibili:
  - `iam:PutGroupPolicy`
  - `iam:PutRolePolicy`
  - `iam:PutUserPolicy`
  - `iam:CreatePolicy`
  - `iam:CreatePolicyVersion`
  - `iam:AttachGroupPolicy`
  - `iam:AttachRolePolicy`
  - `iam:AttachUserPolicy`
- Assicurati che le policy non concedano autorizzazioni per servizi che non utilizzi. Ad esempio, se utilizzi politiche gestite, assicurati che le politiche AWS gestite utilizzate nel tuo account AWS riguardino i servizi che utilizzi effettivamente. Per scoprire quali policy AWS gestite sono in uso nel tuo account, utilizza l'`GetAccountAuthorizationDetails`API IAM (AWS CLI comando: `aws iam get-account-authorization-details`).
- Se la policy concede a un utente l'autorizzazione ad avviare un'istanza di Amazon EC2, potrebbe consentire anche l'operazione `iam:PassRole`, ma in questo caso dovrebbe elencare in modo esplicito i ruoli che l'utente può passare all'istanza di Amazon EC2.
- Esamina tutti i valori per l'elemento `Action` o `Resource` che includono `*`. Quando è possibile, concedi l'accesso `Allow` alle singole operazioni e risorse necessarie agli utenti. Tuttavia, quelle che seguono sono ragioni per cui potrebbe essere utile utilizzare `*` in una policy:
  - La policy è concepita per la concessione di autorizzazioni a livello amministrativo.
  - Per comodità, il carattere jolly viene utilizzato per un set di operazioni simili (ad esempio, `Describe*`): hai così a disposizione l'elenco completo delle operazioni a cui viene fatto riferimento in questo modo.
  - Il carattere jolly viene utilizzato per indicare una classe di risorse o il percorso di una risorsa (ad esempio, `arn:aws:iam::`*`account-id`*`:users/division_abc/*`); puoi concedere l'accesso a tutte le risorse in tale classe o percorso con la massima tranquillità.

  - Un'operazione di servizio non supporta autorizzazioni a livello di risorsa; l'unica scelta per una risorsa è *.
- Esamina i nomi delle policy per assicurarti che riflettano la funzione della policy stessa. Ad esempio, sebbene possa avere un nome che includa la dicitura "di sola lettura", la policy potrebbe effettivamente concedere le autorizzazioni di scrittura o modifica.

Per ulteriori informazioni sulla pianificazione dell'audit di sicurezza, consulta la pagina Best practice per sicurezza, identità e conformità nel Centro di architettura AWS .

Protezione dei dati in AWS Identity and Access Management

Il modello di responsabilità AWS condivisa Modello di di si applica alla protezione dei dati in AWS Identity and Access Management. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con IAM o altri utenti Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia dei dati in IAM e AWS STS

La crittografia dei dati in genere rientra in due categorie: crittografia dei dati a riposo e crittografia dei dati in transito.

Crittografia a riposo

I dati raccolti e archiviati da IAM vengono crittografati quando sono inattivi.

- IAM: i dati raccolti e archiviati all'interno di IAM includono indirizzi IP, metadati dell'account cliente e dati identificativi del cliente, incluse le password. I metadati dell'account cliente e i dati identificativi del cliente vengono crittografati quando sono inattivi utilizzando AES 256 e SHA 256 per gli hash.
- AWS STS— AWS STS non raccoglie i contenuti dei clienti ad eccezione dei registri di servizio che registrano le richieste riuscite, errate e difettose al servizio.

Crittografia in transito

I dati identificativi del cliente, incluse le password, vengono crittografati in transito utilizzando TLS 1.2 e 1.3. Tutti gli AWS STS endpoint supportano HTTPS per la crittografia dei dati in transito. Per un elenco degli AWS STS endpoint, consulta. Regioni ed endpoint

Gestione delle chiavi in IAM e AWS STS

Non è possibile gestire le chiavi di crittografia utilizzando IAM o AWS STS. Per ulteriori informazioni sulle chiavi di crittografia, consulta Cos'è AWS KMS? nella Guida per gli AWS Key Management Service sviluppatori

Privacy del traffico di rete in IAM e AWS STS

Le richieste a IAM devono essere effettuate utilizzando il protocollo TLS (Transport Layer Security Protocol). È possibile proteggere le connessioni al AWS STS servizio utilizzando gli endpoint VPC. Per ulteriori informazioni, consulta Endpoint VPC di interfaccia.

Registrazione e monitoraggio AWS Identity and Access Management

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di AWS Identity and Access Management (IAM), AWS Security Token Service (AWS STS) e AWS delle altre soluzioni. AWS fornisce diversi strumenti per monitorare le AWS risorse e rispondere a potenziali incidenti:

- AWS CloudTrailacquisisce tutte le chiamate API per IAM e AWS STS come eventi, incluse le chiamate dalla console e le chiamate API. Per saperne di più sull'utilizzo CloudTrail con IAM and AWS STS, consultaRegistrazione delle chiamate IAM e AWS STS API con AWS CloudTrail. Per ulteriori informazioni in merito CloudTrail, consulta la Guida AWS CloudTrail per l'utente.
- AWS Identity and Access Management Access Analyzer ti aiuta a identificare le risorse della tua organizzazione e dei tuoi account, come i bucket Amazon S3 o i ruoli IAM, che sono condivisi con un'entità esterna. In questo modo puoi identificare l'accesso non intenzionale alle risorse e ai dati, che rappresenta un rischio per la sicurezza. Per ulteriori informazioni, consulta Cos'è IAM Access Analyzer?
- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Amazon CloudWatch Logs ti aiuta a monitorare, archiviare e accedere ai tuoi file di log da istanze Amazon EC2 e altre CloudTrail fonti. CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.

Per ulteriori risorse e best practice per la sicurezza di IAM, consulta Best practice per la sicurezza e casi d'uso in AWS Identity and Access Management.

Convalida della conformità per AWS Identity and Access Management

I revisori di terze parti valutano la sicurezza e la conformità di AWS Identity and Access Management (IAM) nell'ambito di più programmi di AWS conformità. tra cui SOC, PCI, FedRAMP, ISO e altri.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Ambito per programma di conformitàServizi AWS di conformità e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services: questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee all'HIPAA.

> **Note**
>
> Non Servizi AWS tutte sono idonee all'HIPAA. Per ulteriori informazioni, consulta la sezione Riferimenti sui servizi conformi ai requisiti HIPAA.

- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e

mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).

- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza in AWS Identity and Access Management

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni hanno più zone di disponibilità fisicamente separate e isolate, collegate tramite reti a bassa latenza, ad alto throughput e altamente ridondanti. Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

AWS Identity and Access Management (IAM) e AWS Security Token Service (AWS STS) sono servizi autosufficienti, basati sulla regione, disponibili a livello globale.

IAM è fondamentale. Servizio AWS Ogni operazione eseguita in AWS deve essere autenticata e autorizzata da IAM. IAM verifica ogni richiesta in base alle identità e alle policy archiviate in IAM per determinare se accettare o negare la richiesta. IAM è stato progettato con un piano di controllo e un piano dati separati in modo che il servizio si autentichi anche in caso di errori imprevisti. Le risorse IAM utilizzate nelle autorizzazioni, come ad esempio ruoli e policy, vengono archiviate nel piano di controllo. I clienti IAM possono modificare la configurazione di queste risorse utilizzando operazioni IAM come `DeletePolicy` e `AttachRolePolicy`. Tali richieste di modifica della configurazione pervengono al piano di controllo. Esiste un unico piano di controllo IAM per tutte

le attività commerciali Regioni AWS, che si trova nella regione degli Stati Uniti orientali (Virginia settentrionale). Il sistema IAM propaga quindi le modifiche di configurazione ai piani dati IAM in ogni Regione AWS abilitata. Il piano dati IAM è essenzialmente una replica in sola lettura dei dati di configurazione del piano di controllo IAM. Ciascuno di essi Regione AWS dispone di un'istanza completamente indipendente del piano dati IAM, che esegue l'autenticazione e l'autorizzazione per le richieste provenienti dalla stessa regione. In ogni regione, il piano dati IAM è distribuito su almeno tre zone di disponibilità e ha una capacità sufficiente per tollerare la perdita di una zona di disponibilità senza conseguenze per il cliente. Sia il piano di controllo che il piano dati di IAM sono stati progettati per l'assenza di tempi di inattività pianificati, con tutti gli aggiornamenti software e le operazioni di dimensionamento eseguite in modo impercettibile per i clienti.

AWS STS per impostazione predefinita, le richieste vanno sempre a un singolo endpoint globale. Puoi scegliere di utilizzare un endpoint AWS STS regionale per ridurre la latenza o fornire ridondanza aggiuntiva alle applicazioni. Per ulteriori informazioni, consulta Gestire AWS STS in un Regione AWS.

Alcuni eventi possono interrompere le comunicazioni Regioni AWS attraverso la rete. Tuttavia, anche quando non puoi comunicare con l'endpoint IAM globale, AWS STS puoi comunque autenticare i principi IAM e IAM può autorizzare le tue richieste. I dettagli specifici di un evento che interrompe la comunicazione determineranno la tua capacità di accedere ai servizi. AWS Nella maggior parte delle situazioni, puoi continuare a utilizzare le credenziali IAM nel tuo AWS ambiente. Le seguenti condizioni possono applicarsi a un evento che interrompe la comunicazione.

Chiavi di accesso per gli utenti IAM

Puoi autenticarti a tempo indeterminato in una regione con le chiavi di accesso per gli utenti IAM a lungo termine. Quando utilizzi le API AWS Command Line Interface and, puoi fornire chiavi di AWS accesso per verificare la AWS tua identità nelle richieste programmatiche.

> **Important**
>
> Come best practice, consigliamo che i tuoi utenti eseguano l'accesso con le credenziali temporanee al posto delle chiavi di accesso a lungo termine.

Credenziali temporanee

Puoi richiedere nuove credenziali temporanee con l'endpoint del servizio AWS STS regionale per almeno 24 ore. Le seguenti operazioni API generano credenziali temporanee.

- AssumeRole
- AssumeRoleWithWebIdentity
- AssumeRoleWithSAML
- GetFederationToken
- GetSessionToken

Principali e autorizzazioni

- Potresti non essere in grado di aggiungere, modificare o rimuovere i principali o le autorizzazioni in IAM.
- Le tue credenziali potrebbero non riflettere le modifiche alle autorizzazioni che hai applicato in IAM di recente. Per ulteriori informazioni, consulta Le modifiche che apporto non sono sempre immediatamente visibili.

AWS Management Console

- Potresti essere in grado di utilizzare un endpoint di accesso regionale per accedere alla AWS Management Console come utente IAM. Gli endpoint di accesso regionali hanno il seguente formato URL.

  `https://{Account ID}.signin.aws.amazon.com/console?region={Region}`

  Esempio: https://111122223333.signin.aws.amazon.com/console?region=us-west-2
- Potresti non essere in grado di completare l'autenticazione a più fattori (MFA) Universal 2nd Factor (U2F).

Best practice per la resilienza di IAM

AWS ha integrato la resilienza nelle zone Regioni AWS di disponibilità. Se osservi le seguenti best practice IAM nei sistemi che interagiscono con il tuo ambiente, puoi trarre vantaggio da tale resilienza.

1. Utilizza un endpoint di servizio AWS STS regionale anziché l'endpoint globale predefinito.
2. Verifica la configurazione del tuo ambiente alla ricerca di risorse vitali che creano o modificano abitualmente risorse IAM e prepara una soluzione di fallback che utilizzi le risorse IAM esistenti.

Sicurezza dell'infrastruttura nell' AWS Identity and Access Management

In quanto servizio gestito, AWS Identity and Access Management è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere a IAM attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

IAM è accessibile a livello di programmazione utilizzando le API HTTPS che consentono di inviare richieste HTTPS direttamente al servizio. L'API Query restituisce informazioni riservate, incluse le credenziali di sicurezza. Pertanto, è necessario utilizzare HTTPS con tutte le richieste API. Quando utilizzi le API HTTPS, devi includere il codice per firmare in modo digitale le richieste utilizzando le tue credenziali.

È possibile richiamare queste operazioni API da qualsiasi posizione di rete, ma IAM non supporta le policy di accesso basate sulle risorse che possono includere limitazioni sull'indirizzo IP di origine. È inoltre possibile utilizzare le policy IAM per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) o VPC specifici. In effetti, questo isola l'accesso alla rete a una determinata risorsa IAM solo dal VPC specifico all'interno AWS della rete.

Analisi della configurazione e delle vulnerabilità in AWS Identity and Access Management

AWS gestisce le attività di sicurezza di base come l'applicazione di patch al sistema operativo (OS) guest e al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta le seguenti risorse :

- Modello di responsabilità condivisa
- Amazon Web Services: panoramica dei processi di sicurezza (whitepaper)

Le seguenti risorse riguardano anche la configurazione e l'analisi delle vulnerabilità in AWS Identity and Access Management (IAM):

- Convalida della conformità per AWS Identity and Access Management
- Best practice per la sicurezza e casi d'uso in AWS Identity and Access Management

AWS politiche gestite per AWS Identity and Access Management Access Analyzer

Una policy AWS gestita è una policy autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

ReadOnlyAccesso IAM

Utilizza la policy gestita `IAMReadOnlyAccess` per consentire l'accesso in sola lettura alle risorse IAM. Questa policy concede l'autorizzazione per ottenere ed elencare tutte le risorse IAM. Consente di visualizzare dettagli e i report sulle attività per utenti, gruppi, ruoli, policy, provider di identità e dispositivi MFA. Non include la possibilità di creare o eliminare le risorse o di accedere alle risorse di Sistema di analisi degli accessi AWS IAM. Visualizza la policy per l'elenco completo di servizi e operazioni supportati dalla policy.

UserChangePassword IAM

Utilizza la policy gestita `IAMUserChangePassword` per consentire agli IAM utenti di modificare le loro password.

Configura le Impostazioni dell'account IAM e la Policy sulle password per consentire agli utenti IAM di modificare la password dell'account IAM. Quando consenti questa operazione, IAM allega la policy seguente a ciascun utente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ChangePassword"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM AccessAnalyzer FullAccess

Utilizza la policy `IAMAccessAnalyzerFullAccess` AWS gestita per consentire agli amministratori di accedere a IAM Access Analyzer.

Raggruppamenti di autorizzazioni

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- Sistema di analisi degli accessi AWS IAM: concede le autorizzazioni amministrative complete a tutte le risorse in Sistema di analisi degli accessi AWS IAM.
- Crea ruolo collegato ai servizi: consente all'amministratore di creare un ruolo collegato ai servizi che consente a Sistema di analisi degli accessi AWS IAM di analizzare le risorse in altri servizi per tuo conto. Questa autorizzazione consente di creare il ruolo collegato ai servizi solo per l'utilizzo da parte di Sistema di analisi degli accessi AWS IAM.
- AWS Organizations: consente agli amministratori di utilizzare Sistema di analisi degli accessi AWS IAM per un'organizzazione in AWS Organizations. Dopo aver abilitato l'accesso affidabile per IAM Access Analyzer in AWS Organizations, i membri dell'account di gestione possono visualizzare i risultati in tutta l'organizzazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "access-analyzer:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "access-analyzer.amazonaws.com"
        }
      }
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListChildren",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListRoots"
      ],
      "Resource": "*"
    }
  ]
}
```

Accesso IAM AccessAnalyzer ReadOnly

Utilizza la policy `IAMAccessAnalyzerReadOnlyAccess` AWS gestita per consentire l'accesso in sola lettura a IAM Access Analyzer.

Per consentire anche l'accesso in sola lettura a IAM Access Analyzer per AWS Organizations, crea una policy gestita dal cliente che consenta le azioni Descrivi ed Elenca dalla policy gestita. IAM AccessAnalyzer FullAccess AWS

Autorizzazioni a livello di servizio

Questa policy fornisce accesso in sola lettura a Sistema di analisi degli accessi AWS IAM. In questa policy non sono incluse altre autorizzazioni di servizio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAccessAnalyzerReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "access-analyzer:CheckAccessNotGranted",
```

```
        "access-analyzer:CheckNoNewAccess",
        "access-analyzer:Get*",
        "access-analyzer:List*",
        "access-analyzer:ValidatePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

AccessAnalyzerServiceRolePolitica

Non puoi collegarti AccessAnalyzerServiceRolePolicy alle tue entità IAM. Questa policy è collegata a un ruolo collegato ai servizi che consente a Sistema di analisi degli accessi AWS IAM di eseguire operazioni per tuo conto. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer.

Raggruppamenti di autorizzazioni

Questa policy consente l'accesso a Sistema di analisi degli accessi AWS IAM per analizzare i metadati delle risorse da più Servizi AWS.

- Amazon DynamoDB: consente le autorizzazioni per visualizzare flussi e tabelle DynamoDB.
- Amazon Elastic Compute Cloud: consente le autorizzazioni per descrivere indirizzi IP, istantanee e VPC.
- Amazon Elastic Container Registry: consente le autorizzazioni per descrivere i repository di immagini e recuperare le policy dei repository.
- Amazon Elastic File System: consente le autorizzazioni per visualizzare la descrizione di un file system Amazon EFS e visualizzare la policy a livello di risorse per un file system Amazon EFS.
- AWS Identity and Access Management— Consente le autorizzazioni per recuperare informazioni su un ruolo specificato ed elencare i ruoli IAM con un prefisso di percorso specificato. Consente alle autorizzazioni di recuperare informazioni su utenti, gruppi di utenti, profili di accesso, chiavi di accesso e dati dell'ultimo accesso al servizio.
- AWS Key Management Service— Consente le autorizzazioni per visualizzare informazioni dettagliate su una chiave KMS e sulle relative policy e concessioni chiave.
- AWS Lambda— Consente le autorizzazioni per visualizzare informazioni su alias, funzioni, livelli e alias Lambda.

- AWS Organizations— Concede autorizzazioni alle Organizzazioni e consente la creazione di un analizzatore all'interno dell' AWS organizzazione come zona di fiducia.
- Amazon Relational Database Service: consente le autorizzazioni per visualizzare informazioni dettagliate sugli snapshot del database Amazon RDS e sugli snapshot dei cluster di database Amazon RDS.
- Amazon Simple Storage Service: consente le autorizzazioni per visualizzare informazioni dettagliate sui punti di accesso, i bucket e i bucket di directory Amazon S3 che utilizzano la classe di storage Amazon S3 Express One.
- AWS Secrets Manager— Consente le autorizzazioni per visualizzare informazioni dettagliate sui segreti e sulle policy delle risorse allegate ai segreti.
- Amazon Simple Notification Service: consente le autorizzazioni per visualizzare informazioni dettagliate su un argomento.
- Amazon Simple Queue Service: consente le autorizzazioni per visualizzare informazioni dettagliate sulle code specificate.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessAnalyzerServiceRolePolicy",
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetResourcePolicy",
        "dynamodb:ListStreams",
        "dynamodb:ListTables",
        "ec2:DescribeAddresses",
        "ec2:DescribeByoipCidrs",
        "ec2:DescribeSnapshotAttribute",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcs",
        "ec2:GetSnapshotBlockPublicAccessState",
        "ecr:DescribeRepositories",
        "ecr:GetRepositoryPolicy",
        "elasticfilesystem:DescribeFileSystemPolicy",
        "elasticfilesystem:DescribeFileSystems",
        "iam:GetRole",
        "iam:ListEntitiesForPolicy",
        "iam:ListRoles",
```

```
        "iam:ListUsers",
        "iam:GetUser",
        "iam:GetGroup",
        "iam:GenerateServiceLastAccessedDetails",
        "iam:GetServiceLastAccessedDetails",
        "iam:ListAccessKeys",
        "iam:GetLoginProfile",
        "iam:GetAccessKeyLastUsed",
        "iam:ListRolePolicies",
        "iam:GetRolePolicy",
        "iam:ListAttachedRolePolicies",
        "iam:ListUserPolicies",
        "iam:GetUserPolicy",
        "iam:ListAttachedUserPolicies",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:ListGroupsForUser",
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:ListGrants",
        "kms:ListKeyPolicies",
        "kms:ListKeys",
        "lambda:GetFunctionUrlConfig",
        "lambda:GetLayerVersionPolicy",
        "lambda:GetPolicy",
        "lambda:ListAliases",
        "lambda:ListFunctions",
        "lambda:ListLayers",
        "lambda:ListLayerVersions",
        "lambda:ListVersionsByFunction",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:ListChildren",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListRoots",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBSnapshotAttributes",
```

```
        "rds:DescribeDBSnapshots",
        "s3:DescribeMultiRegionAccessPointOperation",
        "s3:GetAccessPoint",
        "s3:GetAccessPointPolicy",
        "s3:GetAccessPointPolicyStatus",
        "s3:GetAccountPublicAccessBlock",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:GetBucketPolicyStatus",
        "s3:GetBucketPolicy",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetMultiRegionAccessPoint",
        "s3:GetMultiRegionAccessPointPolicy",
        "s3:GetMultiRegionAccessPointPolicyStatus",
        "s3:ListAccessPoints",
        "s3:ListAllMyBuckets",
        "s3:ListMultiRegionAccessPoints",
        "s3express:GetBucketPolicy",
        "s3express:ListAllMyDirectoryBuckets",
        "sns:GetTopicAttributes",
        "sns:ListTopics",
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:ListSecrets",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues"
      ],
      "Resource": "*"
    }
  ]
}
```

Aggiornamenti di IAM e IAM alle policy gestite da AWS

Visualizza i dettagli sugli aggiornamenti a IAM e alle policy AWS gestite da quando il servizio ha iniziato a tracciare queste modifiche. Per ricevere gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS su IAM e nelle pagine della cronologia di Sistema di analisi degli accessi AWS IAM.

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per l'autorizzazione a recuperare informazioni sulle politiche di utenti e ruoli IAM alle autorizzazioni a livello di servizio di. `AccessAnalyzerServiceRolePolicy`	30 maggio 2024
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per l'autorizzazione a recuperare lo stato corrente dell'accesso pubblico a blocchi per gli snapshot di Amazon EC2 alle autorizzazioni a livello di servizio di. `AccessAnalyzerServiceRolePolicy`	23 gennaio 2024
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per i flussi e le tabelle DynamoDB alle autorizzazioni a livello di servizio di. `AccessAnalyzerServiceRolePolicy`	11 gennaio 2024
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per i bucket di directory Amazon S3 alle autorizzazioni a livello di servizio di. `AccessAnalyzerServiceRolePolicy`	1 dicembre 2023

Modifica	Descrizione	Data
IAM AccessAnalyzer ReadOnly Access: autorizzazioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto le autorizzazioni per consentirti di verificare se gli aggiornamenti alle tue policy garantiscono un accesso aggiuntivo. Questa autorizzazione è richiesta da Sistema di analisi degli accessi AWS IAM per eseguire i controlli delle policy sulla policy.	26 novembre 2023
AccessAnalyzerServiceRolePolicy: autorizzazioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto le operazioni IAM alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` per supportare le seguenti operazioni: • Elencare le entità per una policy • Generare dettagli sull'ultimo accesso al servizio • Elencare le informazioni sulla chiave di accesso	26 novembre 2023

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto il supporto per i seguenti tipi di risorse alle autorizzazioni a livello di servizio di`AccessAnalyzerServiceRolePolicy` : • Snapshot del volume Amazon EBS • Repository di Amazon ECR • File system di Amazon EFS • Snapshot del database Amazon RDS • Snapshot del cluster database Amazon RDS • Argomenti di Amazon SNS	25 ottobre 2022
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto l'operazione `lambda:GetFunctionUrlConfig` alle autorizzazioni a livello di servizio di `AccessAnalyzerServiceRolePolicy` .	6 aprile 2022
AccessAnalyzerServiceRolePolitica: autorizzazioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto nuove operazioni di Amazon S3 per analizzare i metadati associati ai punti di accesso multi-regione.	2 settembre 2021

Modifica	Descrizione	Data
IAM AccessAnalyzer ReadOnly Access: autorizza zioni aggiunte	Sistema di analisi degli accessi AWS IAM ha aggiunto una nuova operazione per concedere le autorizzazioni `ValidatePolicy` per consentire all'utente di utilizzar e i controlli delle policy per la convalida. Questa autorizzazione è richiesta da Sistema di analisi degli accessi AWS IAM per eseguire i controlli delle policy sulla policy.	16 marzo 2021
Sistema di analisi degli accessi AWS IAM ha iniziato il monitoraggio delle modifiche	IAM Access Analyzer ha iniziato a tracciare le modifiche per le sue policy AWS gestite.	1 marzo 2021

Usando AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer offre le seguenti funzionalità:

- Gli analizzatori degli accessi esterni di Sistema di analisi degli accessi AWS IAM consente di identificare le risorse nell'organizzazione e negli account che sono condivise con un'entità esterna.
- Gli analizzatori di accessi inutilizzati di Sistema di analisi degli accessi AWS IAM aiutano a identificare gli accessi inutilizzati nell'organizzazione e negli account.
- IAM Access Analyzer convalida le policy IAM rispetto alla grammatica e AWS alle best practice delle policy.
- I controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM a convalidare le policy IAM rispetto agli standard di sicurezza specificati.
- IAM Access Analyzer genera policy IAM basate sull'attività di accesso nei log. AWS CloudTrail

Identificazione delle risorse condivise con un'entità esterna

Sistema di analisi degli accessi AWS IAM consente di identificare le risorse nell'organizzazione e negli account, ad esempio bucket Amazon S3 o ruoli IAM, condivise con un'entità esterna. In questo modo puoi identificare l'accesso non intenzionale alle risorse e ai dati, che rappresenta un rischio per la sicurezza. IAM Access Analyzer identifica le risorse condivise con responsabili esterni utilizzando il ragionamento basato sulla logica per analizzare le politiche basate sulle risorse nel tuo ambiente. AWS Per ogni istanza di una risorsa condivisa al di fuori dell'account, Sistema di analisi degli accessi AWS IAM genera un risultato. I risultati comprendono informazioni sull'accesso e sull'entità esterna a cui è concesso. Puoi rivedere i risultati per determinare se l'accesso è intenzionale e sicuro o se è involontario e rappresenta un rischio per la sicurezza. Oltre a facilitare l'identificazione delle risorse condivise con un'entità esterna, puoi utilizzare i risultati di Sistema di analisi degli accessi AWS IAM per visualizzare in anteprima il modo in cui le policy influiscono sull'accesso multi-account e pubblico sulla risorsa prima di implementare le autorizzazioni delle risorse. I risultati sono organizzati in una dashboard visiva riassuntiva. La dashboard evidenzia la suddivisione tra risultati relativi all'accesso multi-account e pubblico e differenzia i risultati per tipo di risorsa. Per ulteriori informazioni sulle dashboard, consulta Visualizzazione della dashboard dei risultati di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Un'entità esterna può essere un altro AWS account, un utente root, un utente o ruolo IAM, un utente federato, un AWS servizio, un utente anonimo o un'altra entità che puoi utilizzare per creare un filtro. For more information, see Elementi della policy JSON di AWS : entità principale.

Quando abiliti Sistema di analisi degli accessi AWS IAM, crei un analizzatore per l'intera organizzazione o per il tuo account. L'organizzazione o l'account scelto è noto come zona di attendibilità per l'analizzatore. L'analizzatore monitora tutte le risorse supportate all'interno della zona di attendibilità. È considerato attendibile qualsiasi accesso alle risorse da parte delle entità principali che si trovano all'interno della zona di attendibilità. Una volta abilitato, Sistema di analisi degli accessi AWS IAM analizza le policy applicate a tutte le risorse supportate nella zona di attendibilità. Dopo la prima analisi, Sistema di analisi degli accessi AWS IAM analizza queste policy periodicamente. Se aggiungi una nuova policy o ne modifichi una esistente, Sistema di analisi degli accessi AWS IAM analizza la policy nuova o aggiornata entro circa 30 minuti.

Durante l'analisi delle policy, se Sistema di analisi degli accessi AWS IAM ne identifica una che concede l'accesso a un principale esterno che non rientra nella zona di attendibilità, viene generato un risultato. Ogni risultato include i dettagli sulla risorsa, sull'entità esterna che ha accesso e sulle autorizzazioni concesse in modo da poter intraprendere le azioni appropriate. Puoi visualizzare i dettagli inclusi nel risultato per determinare se l'accesso alla risorsa è intenzionale o un potenziale rischio da risolvere. Quando aggiungi una policy a una risorsa o aggiorni una policy esistente, per prima cosa Sistema di analisi degli accessi AWS IAM la analizza. Sistema di analisi degli accessi AWS IAM inoltre analizza periodicamente tutte le policy basate sulle risorse.

In rare occasioni, in determinate condizioni, IAM Access Analyzer non riceve la notifica di una policy aggiunta o aggiornata, il che può causare ritardi nella generazione dei risultati. IAM Access Analyzer può impiegare fino a 6 ore per generare o risolvere i risultati se crei o elimini un punto di accesso multiregionale associato a un bucket Amazon S3 o aggiorni la policy per il punto di accesso multiregionale. Inoltre, se si verifica un problema di consegna con la consegna dei AWS CloudTrail log, la modifica della politica non comporta una nuova scansione della risorsa riportata nel risultato. In questo caso, Sistema di analisi degli accessi AWS IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva, che avviene entro 24 ore. Se desideri confermare che una modifica apportata a una policy risolve un problema di accesso segnalato in un risultato, puoi eseguire nuovamente la scansione della risorsa segnalata in un risultato utilizzando il collegamento Rescan (Nuova scansione) nella pagina dei dettagli Findings (Risultati) o utilizzando

l'operazione `StartResourceScan` dell'API di Sistema di analisi degli accessi AWS IAM. Per ulteriori informazioni, consulta Risoluzione dei risultati.

> **Important**
>
> IAM Access Analyzer analizza solo le politiche applicate alle risorse nella stessa AWS regione in cui è abilitato. Per monitorare tutte le risorse del tuo AWS ambiente, devi creare un analizzatore per abilitare IAM Access Analyzer in ogni regione in cui utilizzi risorse supportate. AWS

Sistema di analisi degli accessi AWS IAM analizza i seguenti tipi di risorse:

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- AWS Identity and Access Management ruoli
- AWS Key Management Service chiavi
- AWS Lambda funzioni e livelli
- Code Amazon Simple Queue Service
- AWS Secrets Manager segreti
- Argomenti su Amazon Simple Notification Service
- Volumi e snapshot di Amazon Elastic Block Store
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service
- Repository di Amazon Elastic Container Registry
- File system di Amazon Elastic File System
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Identificazione dell'accesso inutilizzato concesso a utenti e ruoli IAM

IAM Access Analyzer ti aiuta a identificare e rivedere gli accessi non utilizzati nella tua organizzazione e nei tuoi AWS account. Sistema di analisi degli accessi AWS IAM monitora

continuamente tutti i ruoli e gli utenti IAM dell'organizzazione e degli account AWS e genera risultati per gli accessi inutilizzati. I risultati evidenziano ruoli inutilizzati, chiavi di accesso inutilizzate per gli utenti IAM e password inutilizzate per gli utenti IAM. Per i ruoli e gli utenti IAM attivi, i risultati forniscono visibilità su servizi e operazioni inutilizzati.

I risultati relativi agli analizzatori degli accessi esterni e di quelli inutilizzati sono organizzati in una dashboard visiva riassuntiva. La dashboard evidenzia i risultati Account AWS che hanno ottenuto il maggior numero di risultati e fornisce una suddivisione dei risultati per tipo. Per ulteriori informazioni sulle pagine del pannello di controllo, consulta Visualizzazione della dashboard dei risultati di Sistema di analisi degli accessi AWS IAM.

IAM Access Analyzer esamina le ultime informazioni a cui si accede per tutti i ruoli AWS dell'organizzazione e gli account per aiutarti a identificare gli accessi non utilizzati. Le ultime informazioni a cui si è effettuato l'accesso per le operazioni IAM aiutano a identificare le azioni inutilizzate per i ruoli all'interno degli Account AWS. Per ulteriori informazioni, consulta Perfezionamento delle autorizzazioni per AWS l'utilizzo delle informazioni dell'ultimo accesso.

Convalida delle policy rispetto alle best practices AWS

È possibile convalidare le policy in rapporto alla sintassi della policy IAM e alle best practice AWS utilizzando i controlli delle policy di base forniti dalla convalida delle policy di Sistema di analisi degli accessi AWS IAM. Puoi creare o modificare una policy utilizzando l' AWS API o AWS CLI l'editor di policy JSON nella console IAM. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono consigli pratici che ti aiutano a creare policy funzionali e conformi alle migliori pratiche. AWS Per ulteriori informazioni sulla convalida delle policy tramite l'apposito procedimento, consulta Convalida delle policy di Sistema di analisi degli accessi AWS IAM.

Convalida delle policy rispetto agli standard di sicurezza specificati

È possibile convalidare le policy rispetto agli standard di sicurezza specificati utilizzando i controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM. Puoi creare o modificare una policy utilizzando l' AWS API o AWS CLI l'editor di policy JSON nella console IAM. Tramite la console, puoi verificare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente. AWS CLI Tramite un' AWS API, puoi anche verificare che azioni IAM specifiche che ritieni critiche non siano consentite da una policy. Questi controlli evidenziano un'istruzione di policy che concede nuovi accessi. È possibile aggiornare l'istruzione di policy ed eseguire nuovamente i controlli

finché la policy non sarà conforme allo standard di sicurezza. Per ulteriori informazioni sulla convalida delle policy tramite i controlli delle policy personalizzati, consulta Controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM.

Generazione delle policy

IAM Access Analyzer analizza AWS CloudTrail i log per identificare le azioni e i servizi che sono stati utilizzati da un'entità IAM (utente o ruolo) entro l'intervallo di date specificato. Viene quindi generata una policy IAM basata su tale attività di accesso. È possibile utilizzare la policy generata per perfezionare le autorizzazioni di un'entità collegandola a un utente o ruolo IAM. Per ulteriori informazioni sulla generazione di policy tramite Sistema di analisi degli accessi AWS IAM, consulta Generazione di policy per Sistema di analisi degli accessi AWS IAM.

Prezzi per Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore.

- Verrà addebitato il rispettivo costo per ogni analizzatore degli accessi creato.
- La creazione di analizzatori degli accessi inutilizzati in più regioni comporterà un addebito per ogni analizzatore.
- I ruoli collegati a servizi non vengono analizzati per attività di accesso non utilizzate e non sono inclusi nel numero totale di ruoli IAM analizzati.

Sistema di analisi degli accessi AWS IAM addebita i costi per i controlli delle policy personalizzati in base al numero di richieste API ricevute per verificare la presenza di nuovi accessi.

Per un elenco completo delle tariffe e dei prezzi specifici per Sistema di analisi degli accessi AWS IAM, consulta la pagina dedicata.

Per vedere la tua fattura, vai sul Pannello di controllo di gestione dei costi e della fatturazione nella console AWS Billing and Cost Management. La fattura contiene collegamenti per passare ai report di utilizzo, che consentono di visualizzare i dettagli della fattura. Per ulteriori informazioni sulla Account AWS fatturazione, consulta la Guida per l'utente AWS Billing

Se hai domande sulla AWS fatturazione, sugli account e sugli eventi, contatta. AWS Support

Risultati relativi agli accessi esterni e inutilizzati

Sistema di analisi degli accessi AWS IAM genera risultati per gli accessi esterni e gli accessi inutilizzati nell'organizzazione o nell' Account AWS . Per gli accessi esterni, Sistema di analisi degli accessi AWS IAM genera un risultato per ogni istanza di una policy basata sulle risorse che concede l'accesso a una risorsa nella zona di attendibilità a un principale esterno a tale zona. Quando crei un analizzatore di accessi esterno, scegli un'organizzazione o Account AWS un'analisi. Qualsiasi entità principale nell'organizzazione o nell'account scelto per l'analizzatore viene considerata attendibile. Poiché le entità principali nella stessa organizzazione o account sono attendibili, le risorse e le entità principali all'interno dell'organizzazione o dell'account rappresentano la zona di attendibilità per l'analizzatore. Qualsiasi condivisione all'interno della zona di attendibilità è considerata sicura, quindi Sistema di analisi degli accessi AWS IAM non genera alcun risultato. Ad esempio, se si seleziona un'organizzazione come zona di attendibilità per un analizzatore, tutte le risorse e le entità principali dell'organizzazione si trovano all'interno della zona di attendibilità. Se concedi le autorizzazioni per un bucket Amazon S3 in uno degli account membri dell'organizzazione a un principale in un altro account membro dell'organizzazione, Sistema di analisi degli accessi AWS IAM non genera un risultato. Invece, se concedi l'autorizzazione a un principale in un account che non è membro dell'organizzazione, Sistema di analisi degli accessi AWS IAM genera un risultato.

IAM Access Analyzer genera anche i risultati degli accessi non utilizzati concessi all' AWS organizzazione e agli account. Quando crei un analizzatore di accessi inutilizzato, IAM Access Analyzer monitora continuamente tutti i ruoli e gli utenti IAM nell' AWS organizzazione e negli account e genera risultati sugli accessi non utilizzati. Sistema di analisi degli accessi AWS IAM genera i seguenti tipi di risultati per gli accessi inutilizzati:

- Ruoli inutilizzati: ruoli senza attività di accesso all'interno della finestra di utilizzo specificata.
- Chiavi di accesso e password utente IAM inutilizzate: credenziali appartenenti agli utenti IAM che consentono loro di accedere all' Account AWS.
- Autorizzazioni inutilizzate: autorizzazioni a livello di servizio e a livello di operazione che non sono state utilizzate da un ruolo all'interno della finestra di utilizzo specificata. Sistema di analisi degli accessi AWS IAM utilizza policy basate sull'identità associate ai ruoli per determinare i servizi e le operazioni a cui tali ruoli possono accedere. Sistema di analisi degli accessi AWS IAM supporta la revisione delle autorizzazioni inutilizzate per tutte le autorizzazioni a livello di servizio. Per un elenco completo delle autorizzazioni a livello di operazione supportate per i risultati di accesso inutilizzati, consulta Servizi e operazioni per le informazioni relative all'ultimo accesso a un'operazione IAM.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM fornisce gratuitamente i risultati degli accessi esterni e addebita i costi per i risultati degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Argomenti

Come funzionano i risultati di Sistema di analisi degli accessi AWS IAM

Questo argomento descrive i concetti e i termini utilizzati in IAM Access Analyzer per aiutarti a familiarizzare con il modo in cui IAM Access Analyzer monitora l'accesso alle tue risorse. AWS

Accessi esterni

Per gli analizzatori di accesso esterni, AWS Identity and Access Management Access Analyzer è basato su Zelkova, che traduce le politiche IAM in istruzioni logiche equivalenti ed esegue una suite di risolutori logici generici e specializzati (teorie dei moduli di soddisfacibilità) per risolvere il problema. Sistema di analisi degli accessi AWS IAM applica Zelkova ripetutamente a una policy con query sempre più specifiche per caratterizzare classi di comportamenti consentite dalla policy, in base al contenuto della policy stessa. Per ulteriori informazioni sulle teorie dei moduli di soddisfacibilità, consulta Teorie dei moduli di soddisfacibilità.

Per gli analizzatori degli accessi esterni, Sistema di analisi degli accessi AWS IAM non esamina i log di accesso per determinare se un'entità esterna accede a una risorsa all'interno della zona di attendibilità. Genera un risultato quando una policy basata sulle risorse consente l'accesso a una risorsa, anche se l'entità esterna non ha eseguito l'accesso alla risorsa. Inoltre, Sistema di analisi degli accessi AWS IAM non considera lo stato di alcun account esterno al momento della sua determinazione. In altre parole, se indica che l'account 111122223333 può accedere al bucket Amazon S3, non conosce lo stato degli utenti, dei ruoli, delle policy di controllo del servizio e di altre configurazioni pertinenti in tale account. Questo è per la privacy del cliente: Sistema di analisi degli accessi AWS IAM non considera il proprietario dell'altro account. È anche per la sicurezza: se l'account non è di proprietà del cliente di Sistema di analisi degli accessi AWS IAM, è comunque importante sapere che un'entità esterna potrebbe ottenere l'accesso alle risorse anche se attualmente non ci sono entità nell'account che potrebbero accedere alle risorse.

Sistema di analisi degli accessi AWS IAM considera solo alcune chiavi di condizione IAM che gli utenti esterni non possono influenzare direttamente o che hanno un impatto sull'autorizzazione. Per esempi di chiavi di condizione considerate da Sistema di analisi degli accessi AWS IAM, consulta Chiavi di filtro di Sistema di analisi degli accessi AWS IAM.

Attualmente IAM Access Analyzer non riporta i risultati dei responsabili del servizio o degli account di servizio interni. AWS Nei rari casi in cui Sistema di analisi degli accessi AWS IAM non è in grado di determinare completamente se un'istruzione della policy concede l'accesso a un'entità esterna, si sbaglia nel dichiarare un risultato falso positivo. Sistema di analisi degli accessi AWS IAM è progettato per fornire una visione completa della condivisione delle risorse nell'account e si impegna per ridurre al minimo i falsi negativi.

Accessi inutilizzati

È necessario creare un analizzatore per i risultati degli accessi inutilizzati per i propri ruoli, anche se è già stato creato un analizzatore per generare risultati degli accessi esterni per le proprie risorse.

Dopo aver creato l'analizzatore, Sistema di analisi degli accessi AWS IAM esamina l'attività di accesso per identificare gli accessi inutilizzati. IAM Access Analyzer esamina le ultime informazioni a cui si accede per tutti i ruoli, le chiavi di accesso degli utenti e le password degli utenti nell' AWS organizzazione e negli account per aiutarti a identificare gli accessi non utilizzati. Per i ruoli e gli utenti IAM attivi, Sistema di analisi degli accessi AWS IAM utilizza le ultime informazioni del servizio e delle operazioni IAM a cui si è effettuato l'accesso per identificare le autorizzazioni inutilizzate. Puoi utilizzare analizzatori di accesso inutilizzati per scalare il processo di revisione a livello di organizzazione e account. AWS Puoi utilizzare le ultime informazioni a cui è stato effettuato l'accesso per un'analisi più approfondita dei singoli ruoli.

Pannello di riepilogo

Sia per l'accesso esterno che per quello inutilizzato, Sistema di analisi degli accessi AWS IAM organizza i risultati in una dashboard riassuntiva. Per gli accessi esterni, la dashboard riassuntiva evidenzia la suddivisione tra risultati relativi all'accesso multi-account e pubblico e differenzia i risultati per tipo di risorsa. Per gli accessi non utilizzati, la dashboard evidenzia quelli Account AWS che hanno ottenuto il maggior numero di risultati e fornisce una suddivisione dei risultati per tipo. Dopo aver creato un analizzatore per gli accessi esterni o inutilizzati, Sistema di analisi degli accessi AWS IAM aggiunge automaticamente nuovi risultati alla dashboard per quanto riguarda i ruoli con autorizzazioni inutilizzate.

Nozioni di base sui risultati di AWS Identity and Access Management Access Analyzer

Utilizza le informazioni contenute in questo argomento per conoscere i requisiti necessari per l'utilizzo e la gestione AWS Identity and Access Management Access Analyzer e quindi come abilitare IAM Access Analyzer. Per ulteriori informazioni sul ruolo collegato ai servizi per Sistema di analisi degli accessi AWS IAM, consulta Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer.

Autorizzazioni necessarie per utilizzare Sistema di analisi degli accessi AWS IAM

Per configurare correttamente e utilizzare Sistema di analisi degli accessi AWS IAM, all'account che utilizzi devono essere concesse le autorizzazioni necessarie.

AWS politiche gestite per IAM Access Analyzer

AWS Identity and Access Management Access Analyzer fornisce politiche AWS gestite per aiutarti a iniziare rapidamente.

- IAM AccessAnalyzer FullAccess: consente l'accesso completo a IAM Access Analyzer per gli amministratori. Questa policy consente inoltre di creare i ruoli collegati ai servizi necessari per consentire a IAM Access Analyzer di analizzare le risorse dell'account o dell'organizzazione. AWS
- IAM AccessAnalyzer ReadOnly Access: consente l'accesso in sola lettura a IAM Access Analyzer. È necessario aggiungere ulteriori policy alle identità IAM (utenti, gruppi di utenti o ruoli) per consentire loro di visualizzare i risultati.

Risorse definite da Sistema di analisi degli accessi AWS IAM

Per visualizzare le risorse definite da Sistema di analisi degli accessi AWS IAM, consulta Tipi di risorsa definiti da Sistema di analisi degli accessi AWS IAM in Service Authorization Reference.

Autorizzazioni di servizio necessarie per Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM utilizza un ruolo collegato ai servizi (SLR) chiamato `AWSServiceRoleForAccessAnalyzer`. Questa reflex garantisce al servizio l'accesso in sola lettura per analizzare le AWS risorse con policy basate sulle risorse e analizzare gli accessi non utilizzati per conto dell'utente. Il servizio crea il ruolo nel tuo account nei seguenti casi:

- Crei un analizzatore degli accessi esterni con il tuo account come zona di attendibilità.
- Crei un analizzatore degli accessi inutilizzati con il tuo account come account selezionato.

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM è un servizio regionale. Per gli accessi esterni, devi abilitare Sistema di analisi degli accessi AWS IAM in ogni regione in modo indipendente. Per gli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

In alcuni casi, dopo aver creato un analizzatore degli accessi esterni o inutilizzati in Sistema di analisi degli accessi AWS IAM, la pagina Risultati o la dashboard vengono caricate senza risultati o riepiloghi. Ciò potrebbe essere dovuto a un ritardo nella console per la compilazione dei risultati. Potrebbe essere necessario aggiornare manualmente il browser o ricontrollare più tardi per

visualizzare i risultati o il riepilogo. Se ancora non viene visualizzato alcun risultato per l'analizzatore degli accessi esterni, non hai nel tuo account le risorse supportate a cui è possibile accedere da un'entità esterna. Se una policy che concede l'accesso a un'entità esterna viene applicata a una risorsa, Sistema di analisi degli accessi AWS IAM genera un risultato.

> **Note**
>
> Per gli analizzatori degli accessi esterni, potrebbero essere necessari fino a 30 minuti dopo la modifica di una policy perché Sistema di analisi degli accessi AWS IAM possa analizzare la risorsa e generare un nuovo risultato o aggiornarne uno esistente per l'accesso alla risorsa. Sia per gli analizzatori degli accessi esterni che per quelli inutilizzati, gli aggiornamenti dei risultati potrebbero non essere riportati immediatamente nella dashboard.

Autorizzazioni Sistema di analisi degli accessi AWS IAM necessarie per visualizzare la dashboard dei risultati

Per visualizzare la dashboard dei risultati di Sistema di analisi degli accessi AWS IAM, all'account che utilizzi deve essere concesso l'accesso per eseguire le seguenti operazioni necessarie:

- `GetAnalyzer`
- `ListAnalyzers`
- `GetFindingsStatistics`

Per visualizzare le operazioni definite da Sistema di analisi degli accessi AWS IAM, consulta Operazioni definite da Sistema di analisi degli accessi AWS IAM in Service Authorization Reference.

Abilitazione di Sistema di analisi degli accessi AWS IAM

Per creare un analizzatore di accessi esterno con come zona di fiducia Account AWS

Per abilitare un analizzatore degli accessi esterni in una regione, è necessario creare un analizzatore in tale regione. È necessario creare un analizzatore degli accessi esterni in ogni regione in cui desideri monitorare l'accesso alle risorse.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer.
3. Scegli le impostazioni dell'analizzatore.

4. Scegliere Create analyzer (Crea analizzatore).
5. Nella sezione Analisi, scegli Analisi degli accessi esterni.
6. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi AWS IAM.
7. Inserire un nome per l'analizzatore.
8. Scegli Account AWS corrente come zona di attendibilità per l'analizzatore.

> **Note**
>
> Se il tuo account non è l'account di AWS Organizations gestione o l'account di amministratore delegato, puoi creare un solo analizzatore con il tuo account come zona di fiducia.

9. Facoltativo. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Scegli Invia.

Quando crei un analizzatore degli accessi esterni per abilitare Sistema di analisi degli accessi AWS IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Per creare un analizzatore degli accessi esterni con l'organizzazione come zona di attendibilità

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer.
3. Scegli le impostazioni dell'analizzatore.
4. Scegliere Create analyzer (Crea analizzatore).
5. Nella sezione Analisi, scegli Analisi degli accessi esterni.
6. Nella sezione Dettagli dell'analizzatore, verifica che la regione visualizzata sia quella in cui desideri abilitare Sistema di analisi degli accessi AWS IAM.
7. Inserire un nome per l'analizzatore.
8. Scegli Organizzazione attuale come zona di attendibilità per l'analizzatore.
9. Facoltativo. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Scegli Invia.

Quando si crea un analizzatore degli accessi esterni con l'organizzazione come zona di attendibilità, in ogni account dell'organizzazione viene creato un ruolo collegato al servizio denominato `AWSServiceRoleForAccessAnalyzer`.

Per creare un analizzatore degli accessi inutilizzati per l'account corrente

Utilizza la seguente procedura per creare un analizzatore degli accessi inutilizzati per un singolo Account AWS. Per gli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer.
3. Scegli le impostazioni dell'analizzatore.
4. Scegliere Create analyzer (Crea analizzatore).
5. Nella sezione Analisi, scegli Analisi degli accessi inutilizzati.
6. Inserire un nome per l'analizzatore.
7. In Periodo di monitoraggio, inserisci il numero di giorni per i quali generare i risultati delle autorizzazioni inutilizzate. Ad esempio, se inserisci 90 giorni, l'analizzatore genererà i risultati per le entità IAM all'interno dell'account selezionato per tutte le autorizzazioni che non sono state utilizzate da 90 o più giorni dall'ultima scansione. Puoi scegliere un valore compreso tra 1 e 180 giorni.
8. Per Account selezionati, scegli Account AWS corrente.

   > **Note**
   >
   > Se il vostro account non è l'account di AWS Organizations gestione o l'account di amministratore delegato, potete creare un solo analizzatore con il vostro account come account selezionato.

9. Facoltativo. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Scegli Invia.

Quando crei un analizzatore degli accessi inutilizzati per abilitare Sistema di analisi degli accessi AWS IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Per creare un analizzatore degli accessi inutilizzati con l'organizzazione attuale

Utilizzate la seguente procedura per creare un analizzatore di accessi inutilizzato per consentire a un'organizzazione di esaminare centralmente tutto ciò Account AWS che fa parte di un'organizzazione. Per l'analisi degli accessi inutilizzati, i risultati relativi all'analizzatore non cambiano in base alla regione. Non è necessario creare un analizzatore in ogni regione in cui sono disponibili risorse.

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Se un account membro viene rimosso dall'organizzazione, l'analizzatore degli accessi inutilizzati smetterà di generare nuovi risultati e di aggiornare i risultati esistenti per quell'account dopo 24 ore. I risultati associati all'account membro rimosso dall'organizzazione verranno rimossi definitivamente dopo 90 giorni.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer.
3. Scegli le impostazioni dell'analizzatore.
4. Scegliere Create analyzer (Crea analizzatore).
5. Nella sezione Analisi, scegli Analisi degli accessi inutilizzati.
6. Inserire un nome per l'analizzatore.
7. In Periodo di monitoraggio, inserisci il numero di giorni per i quali generare i risultati delle autorizzazioni inutilizzate. Ad esempio, se inserisci 90 giorni, l'analizzatore genererà i risultati per le entità IAM all'interno degli account dell'organizzazione selezionata per tutte le autorizzazioni che non sono state utilizzate da 90 o più giorni dall'ultima scansione dell'analizzatore. Puoi scegliere un valore compreso tra 1 e 180 giorni.
8. Per Account selezionati, scegli Organizzazione attuale come account selezionati per l'analizzatore.

9. Facoltativo. Aggiungere tutti i tag che si desidera applicare all'analizzatore.
10. Scegli Invia.

Quando crei un analizzatore degli accessi inutilizzati per abilitare Sistema di analisi degli accessi AWS IAM, nell'account viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForAccessAnalyzer`.

Stato di Sistema di analisi degli accessi AWS IAM

Per visualizzare lo stato degli analizzatori, scegli Analyzers (Analizzatori). Gli analizzatori creati per un'organizzazione o un account possono avere lo stato seguente:

Stato	Descrizione
Attivo	Per gli analizzatori degli accessi esterni, l'analizzatore monitora attivamente le risorse all'interno della zona di attendibilità. L'analizz atore genera attivamente nuovi risultati e aggiorna quelli esistenti. Per gli analizzatori di accessi non utilizzati, l'analizzatore monitora attivamente gli accessi non utilizzati all'interno dell'organizzazion e selezionata o nel periodo di tracciamento specificato. Account AWS L'analizzatore genera attivamente nuovi risultati e aggiorna quelli esistenti.
Creazione	La creazione dell'analizzatore è ancora in corso. Al termine, l'analizzatore diventa attivo.
Disabilitato	L'analizzatore è disabilitato a causa di un'azione intrapresa dall'amministratore. AWS Organizations ad esempio la rimozione dell'account dell'analizzatore come amministr atore delegato per Sistema di analisi degli accessi AWS IAM. Quando l'analizzatore è in

Stato	Descrizione
	stato disabilitato, non genera nuovi risultati né aggiorna quelli esistenti.
Non riuscito	Creazione dell'analizzatore non è riuscita a causa di un problema di configurazione. L'analizzatore non genererà alcun risultato. Eliminare l'analizzatore e crearne uno nuovo.

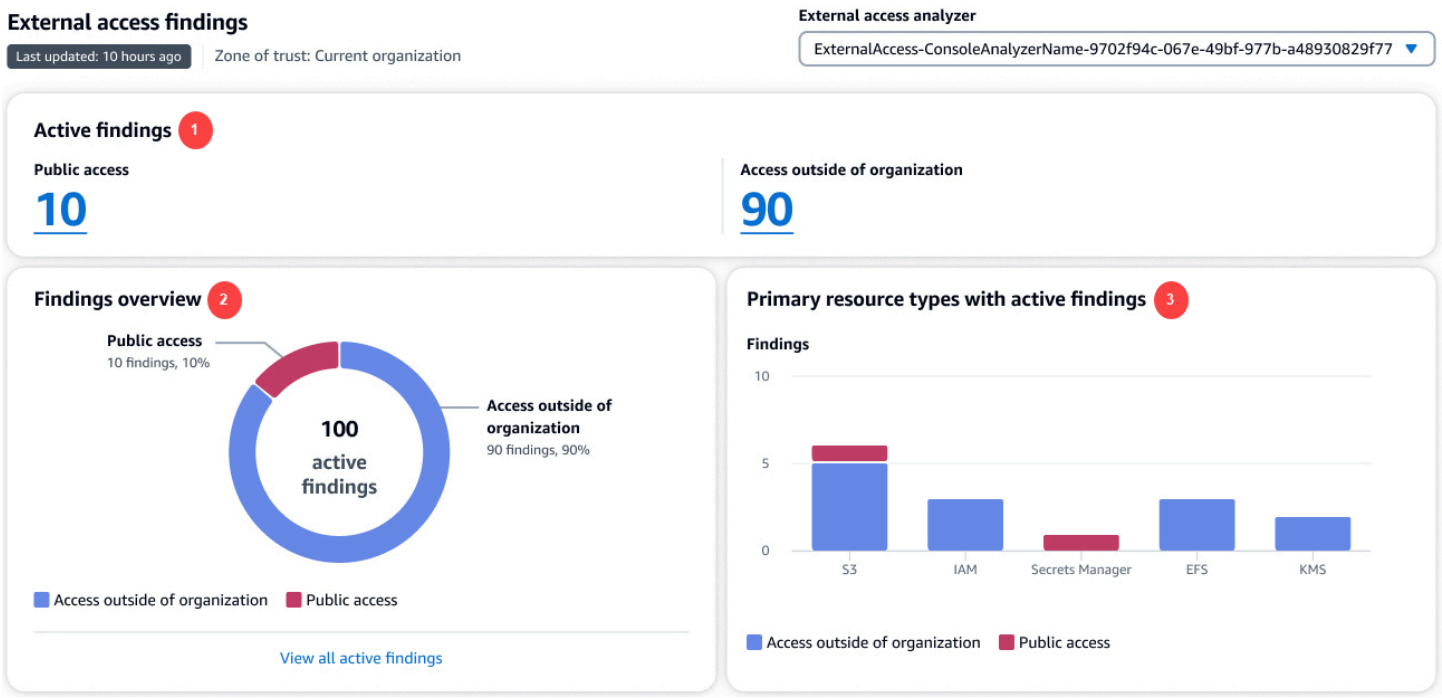
Visualizzazione della dashboard dei risultati di Sistema di analisi degli accessi AWS IAM

AWS Identity and Access Management Access Analyzer organizza i risultati degli accessi esterni e degli accessi non utilizzati in una dashboard riassuntiva visiva. La dashboard ti aiuta a visualizzare in modo completo l'uso efficace delle autorizzazioni su larga scala e a identificare gli account che richiedono attenzione. Puoi utilizzare la dashboard per esaminare i risultati per AWS organizzazione, account e tipo di risultato.

Per visualizzare la dashboard riassuntiva per gli analizzatori degli accessi esterni

> **Note**
>
> Dopo aver creato o aggiornato un analizzatore, può essere necessario del tempo prima che la dashboard riassuntiva rifletta gli aggiornamenti dei risultati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer. Viene visualizzata la finestra Riepilogo.
3. Scegli un analizzatore dal menu a discesa Analizzatore degli accessi esterni. Appare un riepilogo dei risultati dell'analizzatore nella sezione Risultati degli accessi esterni.



Nell'immagine precedente, la dashboard dei risultati degli accessi esterni è visibile nella pagina Riepilogo:

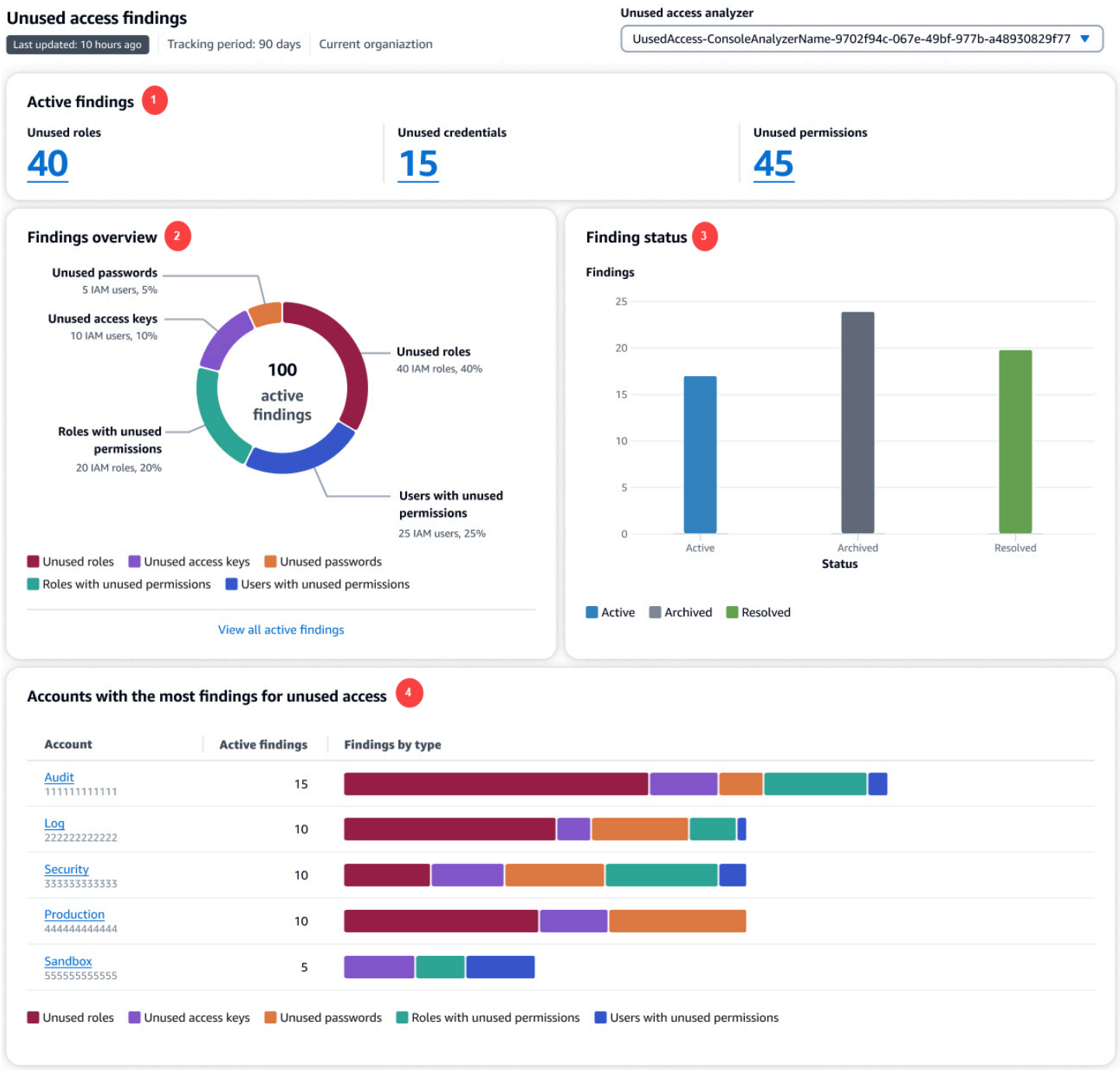
1. La sezione Risultati attivi include il numero di risultati attivi accessibili al pubblico e il numero di risultati attivi che forniscono l'accesso all'esterno dell'account o dell'organizzazione. Scegli un numero per elencare tutti i risultati attivi di ogni tipo.
2. La sezione Panoramica dei risultati include una suddivisione del tipo di risultati attivi. Scegli Visualizza tutti i risultati attivi per ottenere un elenco completo dei risultati attivi per l'account o l'organizzazione dell'analizzatore.
3. La sezione Tipi di risorse primarie con risultati attivi include una suddivisione dei tipi di risorse principali con risultati attivi. Queste informazioni ti aiutano innanzitutto a dare priorità ai risultati per le risorse primarie. Ad esempio, Amazon S3, DynamoDB e. AWS KMS Questo elenco non contiene tutti i tipi di risorse. L'analizzatore potrebbe avere risultati attivi per tipi di risorse non elencati in questa sezione.

Per visualizzare la dashboard riassuntiva per gli analizzatori degli accessi inutilizzati

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Dopo aver creato o aggiornato un analizzatore, in base al numero di utenti e ruoli, può essere necessario del tempo prima che la dashboard riassuntiva rifletta gli aggiornamenti dei risultati.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer. Viene visualizzata la finestra Riepilogo.
3. Scegli un analizzatore dal menu a discesa Analizzatore degli accessi inutilizzati. Appare un riepilogo dei risultati dell'analizzatore nella sezione Risultati degli accessi inutilizzati.



Nell'immagine precedente, la dashboard dei risultati degli accessi esterni è visibile nella pagina Riepilogo:

1. La sezione Risultati attivi include il numero di risultati attivi per ruoli , credenziali e autorizzazioni inutilizzati nell'account o nell'organizzazione. Le credenziali inutilizzate includono i risultati relativi alla chiave di accesso e alle password inutilizzate. Le autorizzazioni inutilizzate includono sia gli

utenti che i ruoli con autorizzazioni inutilizzate. Scegli un numero per elencare tutti i risultati attivi di ogni tipo.

2. La sezione Panoramica dei risultati include una suddivisione del tipo di risultati attivi. Scegli Visualizza tutti i risultati attivi per ottenere un elenco completo dei risultati attivi per l'account o l'organizzazione dell'analizzatore.
3. La sezione Stato dei risultati include un'analisi dettagliata dello stato dei risultati (Attivo, Archiviato e Risolto) per l'account o l'organizzazione.
4. La sezione Account con il numero maggiore di risultati relativi agli accessi inutilizzati viene visualizzata solo se gli account selezionati dell'analizzatore degli accessi inutilizzati sono a livello di organizzazione. Include una suddivisione degli account dell'organizzazione con i risultati più attivi. Questo elenco non contiene tutti gli account dell'organizzazione. L'analizzatore potrebbe avere risultati attivi per altri account non elencati in questa sezione.

Uso dei risultati

Risultati dell'accesso esterno

I risultati degli accessi esterni vengono generati una sola volta per ogni istanza di risorsa condivisa al di fuori della zona di attendibilità. Ogni volta che una policy basata sulle risorse viene modificata, Sistema di analisi degli accessi AWS IAM la rianalizza. Se la policy aggiornata condivide una risorsa già identificata in un risultato, ma con autorizzazioni o condizioni diverse, viene generato un nuovo risultato per l'istanza della condivisione della risorsa. Se l'accesso nel primo risultato viene rimosso, il risultato viene aggiornato nello stato Risolto.

Lo stato di tutti i risultati rimane Attivo fino a quando non vengono archiviati o non si rimuove l'accesso che ha generato il risultato. Quando si rimuove l'accesso, lo stato del risultato viene aggiornato in Risolto.

> **Note**
>
> Dopo la modifica di una policy, perché Sistema di analisi degli accessi AWS IAM possa analizzare la risorsa e aggiornare il risultato degli accessi esterni, potrebbero essere necessari fino a 30 minuti.

Risultati degli accessi inutilizzati

I risultati degli accessi inutilizzati vengono generati per le entità IAM all'interno dell'account o dell'organizzazione selezionati in base al numero di giorni specificato durante la creazione dell'analizzatore. Viene generato un nuovo risultato quando l'analizzatore esegue nuovamente la scansione delle entità se viene soddisfatta una delle seguenti condizioni:

- Un ruolo è inattivo per il numero specificato di giorni.
- Un'autorizzazione, una password utente o una chiave di accesso utente inutilizzate superano il numero di giorni specificato.

Devi esaminare tutti i risultati del tuo account per determinare se l'accesso esterno o inutilizzato è previsto e approvato. Se l'accesso esterno o inutilizzato identificata nel risultato è previsto, è possibile archiviare il risultato. Quando si archivia un risultato, lo stato viene modificato in Archiviato e il risultato viene rimosso dall'elenco dei risultati attivi. Il risultato non viene eliminato. Puoi visualizzare i risultati archiviati in qualsiasi momento. Elabora tutti i risultati nel tuo account fino a quando non hai più risultati attivi. Quando non hai più risultati, sai che eventuali nuovi risultati Attivi generati provengono da una modifica recente dell'ambiente.

> **Note**
>
> I risultati di accesso non utilizzati sono disponibili solo utilizzando l'azione API ListFindingsV2.

Analisi dei risultati

Dopo aver abilitato Sistema di analisi degli accessi AWS IAM, il passaggio successivo consiste nell'esaminare i risultati per determinare se l'accesso identificato nel risultato è intenzionale o meno. Puoi inoltre esaminare i risultati per determinare i risultati simili per l'accesso intenzionale e quindi creare una regola di archiviazione per archiviare tali risultati automaticamente. Puoi esaminare i risultati archiviati e risolti.

Per esaminare i risultati

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegliere Access Analyzer.
3. Viene visualizzata la dashboard dei risultati. Seleziona i risultati attivi per l'analizzatore degli accessi esterni o inutilizzati.

Per ulteriori informazioni sulla visualizzazione della dashboard dei risultati, consulta Visualizzazione della dashboard dei risultati di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> I risultati vengono visualizzati solo se si dispone dell'autorizzazione per visualizzare i risultati per l'analizzatore.

Tutti i risultati vengono visualizzati per l'analizzatore. Per visualizzare altri risultati generati dall'analizzatore, scegli il tipo di risultati dal menu a discesa Stato:

- Scegliere Active (Attivo) per visualizzare tutti i risultati attivi generati dall'analizzatore.
- Scegliere Archived (Archiviato) per visualizzare solo i risultati generati dall'analizzatore che sono stati archiviati. Per ulteriori informazioni, consulta Archiviazione dei risultati.
- Scegliere Resolved (Risolto) per visualizzare solo i risultati generati dall'analizzatore che sono stati risolti. Quando si risolve il problema che ha generato il risultato, lo stato del risultato viene modificato in Risolto.

  > **Important**
  >
  > I risultati risolti vengono eliminati 90 giorni dopo l'ultimo aggiornamento del risultato. I risultati attivi e archiviati non vengono eliminati a meno che non si elimini l'analizzatore che li ha generati.

- Scegliere All (Tutto) per visualizzare tutti i risultati con qualsiasi stato che sono stati generati dall'analizzatore.

Risultati dell'accesso esterno

Scegli Accesso esterno, quindi seleziona l'analizzatore degli accessi esterni dal menu a discesa Visualizza analizzatore. Nella pagina Risultati per gli analizzatori degli accessi esterni vengono visualizzati i seguenti dettagli sull'istruzione della policy e della risorsa condivisa che ha generato il risultato:

ID risultato

L'ID univoco assegnato al risultato. Scegliere l'ID risultato per visualizzare ulteriori dettagli sull'istruzione della policy e della risorsa che ha generato il risultato.

Resource (Risorsa)

Il tipo e il nome parziale della risorsa a cui è applicata una policy che consente l'accesso a un'entità esterna non all'interno della zona di attendibilità.

Account proprietario della risorsa

Questa colonna viene visualizzata solo se si utilizza un'organizzazione come zona di attendibilità. L'account dell'organizzazione proprietaria della risorsa riportata nel risultato.

External principal (Entità principale esterna)

L'entità principale, non all'interno della zona di attendibilità, a cui la policy analizzata concede l'accesso. I valori validi includono:

- Account AWS: tutti i principali riportati nell' Account AWS con autorizzazioni dall'amministratore dell'account possono accedere alla risorsa.
- Qualsiasi principale: tutti i principali di qualsiasi entità Account AWS che soddisfano le condizioni incluse nella colonna Condizioni sono autorizzati ad accedere alla risorsa. Ad esempio, se un VPC è elencato, significa che può accedere alla risorsa qualsiasi entità principale in qualsiasi account che dispone dell'autorizzazione per accedere al VPC elencato.
- Utente canonico: tutti i principali Account AWS con l'ID utente canonico elencato sono autorizzati ad accedere alla risorsa.
- Ruolo IAM: il ruolo IAM elencato dispone dell'autorizzazione per accedere alla risorsa.
- Utente IAM: l'utente IAM elencato dispone dell'autorizzazione per accedere alla risorsa.

Condition

La condizione dell'istruzione della policy che concede l'accesso. Ad esempio, se il campo Condition (Condizione) include Source VPC (VPC di origine), significa che la risorsa viene condivisa con un'entità che ha accesso al VPC elencato. Le condizioni possono essere globali o specifiche del servizio. Le chiavi di condizione globali hanno il prefisso `aws:`.

Shared through (Condiviso tramite)

Il campo Shared through (Condiviso tramite) indica come viene concesso l'accesso che ha generato il risultato. I valori validi includono:

- Policy del bucket: la policy del bucket collegata al bucket Amazon S3.
- Lista di controllo accessi: la lista di controllo accessi (ACL) collegata al bucket Amazon S3.
- Punto di accesso: un punto di accesso o un punto di accesso multi-regione associato al bucket Amazon S3. L'ARN dell'access point viene visualizzato nei dettagli dei risultati.

Livello di accesso

Livello di accesso concesso all'entità esterna dalle operazioni nella policy basata sulle risorse. Visualizza i dettagli del risultato per ulteriori informazioni. I valori del livello di accesso includono quanto segue:

- Elenco: l'autorizzazione a elencare le risorse all'interno del servizio per determinare l'esistenza di un oggetto. Le operazioni con questo livello di accesso possono elencare gli oggetti, ma consentono di visualizzare i contenuti di una risorsa.
- Lettura: l'autorizzazione a leggere ma non a modificare i contenuti e gli attributi delle risorse del servizio.
- Scrittura: l'autorizzazione a creare, eliminare o modificare le risorse del servizio.
- Autorizzazioni: l'autorizzazione a concedere o modificare le autorizzazioni a livello di risorsa nel servizio.
- Aggiunta di tag: l'autorizzazione per eseguire operazioni che modificano solo lo stato dei tag delle risorse.

Aggiornato

Un timestamp per l'aggiornamento più recente dello stato del risultato o l'ora e la data in cui il risultato è stato generato se non sono stati apportati aggiornamenti.

> **Note**
>
> Dopo la modifica di una policy, perché Sistema di analisi degli accessi AWS IAM possa rianalizzare la risorsa e aggiornare il risultato, potrebbero essere necessari fino a 30 minuti.

Stato

Lo stato del risultato: Active (Attivo), Archived (Archiviato) o Resolved (Risolto).

Risultati degli accessi inutilizzati

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Scegli Accesso inutilizzato, quindi seleziona l'analizzatore degli accessi inutilizzati dal menu a discesa Visualizza analizzatore. Nella pagina Risultati per gli analizzatori degli accessi inutilizzati vengono visualizzati i seguenti dettagli sull'entità IAM che ha generato il risultato:

ID risultato

L'ID univoco assegnato al risultato. Scegli l'ID risultato per visualizzare ulteriori dettagli sull'entità IAM che ha generato il risultato.

Tipo di risultato

Il tipo di risultato di accesso inutilizzato: chiave di accesso inutilizzata, password inutilizzata, autorizzazione inutilizzata o ruolo inutilizzato.

Entità IAM

L'entità IAM riportata nel risultato. Può trattarsi di un utente o di un ruolo IAM.

Account AWS ID

Questa colonna viene visualizzata solo se si configura l'analizzatore per tutti gli utenti dell'organizzazione. Account AWS Account AWS Nell'organizzazione proprietaria dell'entità IAM riportata nel risultato.

Ultimo aggiornamento

L'ultima volta che l'entità IAM riportata nel risultato è stata aggiornata o, se non sono stati eseguiti aggiornamenti, quando l'entità è stata creata.

Stato

Lo stato del risultato (Attivo, Archiviato o Risolto).

Filtro dei risultati

Il filtro predefinito per la pagina dei risultati visualizza tutti i risultati. Per visualizzare i risultati attivi, scegli lo stato Attivo dal menu a discesa Stato. Per visualizzare i risultati archiviati, scegli lo stato

Archiviato dal menu a discesa Stato. Quando inizi a utilizzare per la prima volta Sistema di analisi degli accessi AWS IAM, non ci sono risultati archiviati.

Utilizza i filtri per visualizzare solo i risultati che soddisfano i criteri di proprietà specificati. Per creare un filtro, seleziona la proprietà su cui filtrare, quindi scegli se la proprietà è uguale o contiene un valore e seleziona un valore di proprietà su cui filtrare. Ad esempio, per creare un filtro che mostri solo i risultati per uno specifico Account AWS, scegli AWS Account per la proprietà, quindi scegli AWS Account =, quindi inserisci il numero di conto per il Account AWS quale desideri visualizzare i risultati.

Per l'elenco delle chiavi di filtro che puoi utilizzare per creare o aggiornare una regola di archivio, consulta Chiavi di filtro di Sistema di analisi degli accessi AWS IAM.

Filtraggio dei risultati degli accessi esterni

Per filtrare i risultati degli accessi esterni

1. Scegli Accesso esterno, quindi seleziona l'analizzatore dal menu a discesa Visualizza analizzatore.
2. Scegli la casella di ricerca per visualizzare un elenco di proprietà disponibili.
3. Scegliere la proprietà da utilizzare per filtrare i risultati visualizzati.
4. Scegliere il valore da corrispondere per la proprietà. Vengono visualizzati solo i risultati con tale valore presente nel risultato.

   Ad esempio, scegli Risorsa come proprietà e poi Risorsa:, digita il nome parziale o completo di un bucket, quindi premi Invio. Vengono visualizzati solo i risultati per il bucket che corrisponde ai criteri del filtro. Per creare un filtro che visualizzi solo i risultati per le risorse che consentono l'accesso pubblico, puoi scegliere la proprietà Accesso pubblico, quindi selezionare Accesso pubblico = e poi scegliere Accesso pubblico = true.

Puoi aggiungere altre proprietà per filtrare ulteriormente i risultati visualizzati. Quando aggiungi altre proprietà, vengono visualizzati solo i risultati che corrispondono a tutte le condizioni del filtro. La definizione di un filtro per visualizzare i risultati che corrispondono a una proprietà O a un'altra proprietà non è supportata. Scegli Cancella filtri per cancellare tutti i filtri definiti e visualizzare tutti i risultati con lo stato specificato per l'analizzatore.

Alcuni campi sono visibili solo quando si visualizzano i risultati di un analizzatore con un'organizzazione come zona di attendibilità.

Per la definizione dei filtri sono disponibili le seguenti proprietà:

- Accesso pubblico: per filtrare in base ai risultati delle risorse che consentono l'accesso pubblico, usa il filtro Accesso pubblico, quindi scegli Accesso pubblico: true.
- Risorsa: per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa.
- Tipo di risorsa: per filtrare in base al tipo di risorsa, scegli il tipo dall'elenco visualizzato.
- Account del proprietario della risorsa: utilizza questa proprietà per filtrare in base all'account dell'organizzazione proprietaria della risorsa riportata nel risultato.
- AWS Account: utilizza questa proprietà per filtrare Account AWS in base all'accesso concesso nella sezione Principal di una dichiarazione politica. Per filtrare in base Account AWS, digita tutto o parte dell' Account AWS ID a 12 cifre oppure tutto o parte dell'ARN completo dell'account dell' AWS utente o del ruolo esterno che ha accesso alle risorse nell'account corrente.
- Utente canonico: per filtrare in base all'utente canonico, digita l'ID utente canonico come definito per i bucket Amazon S3. Per ulteriori informazioni, consulta ID account di AWS.
- Utente federato: per filtrare in base all'utente federato, digita l'ARN parziale o completo dell'identità federata. Per ulteriori informazioni, consulta Provider di identità e federazione.
- ID risultato: per filtrare in base all'ID risultato, digita l'ID parziale o completo.
- ARN principale: utilizza questa proprietà per filtrare l'ARN del principale (utente, ruolo o gruppo IAM) utilizzato in una chiave di condizione aws:. PrincipalArn Per filtrare in base all'ARN principale, digita tutto o parte dell'ARN dell'utente, del ruolo o del gruppo IAM da un report esterno Account AWS riportato in un risultato.
- OrgID principale: per filtrare in base all'OrgID del principale, digita l'ID organizzazione parziale o completo associato ai principali esterni che appartengono all'organizzazione AWS specificata come condizione nel risultato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Principal OrgPaths: per filtrare per Principal OrgPaths, digita tutto o parte dell'ID AWS dell'organizzazione o dell'unità organizzativa (OU) che consente l'accesso a tutti i responsabili esterni che sono membri dell'account dell'organizzazione o dell'unità organizzativa specificata come condizione nella politica. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Account di origine: per filtrare in base all'account di origine, digita tutto o parte dell' Account AWS ID associato alle risorse, come utilizzato in alcune autorizzazioni interservizi di. AWS Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.

- ARN di origine: per filtrare in base all'ARN di origine, digita l'ARN parziale o completo specificato come condizione nel risultato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- IP di origine: per filtrare in base all'IP di origine, digita l'indirizzo IP parziale o completo che consente alle entità esterne di accedere alle risorse nell'account corrente quando utilizza l'indirizzo IP specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- VPC di origine: per filtrare in base al VPC di origine, digita l'ID parziale o completo del VPC che consente alle entità esterne di accedere alle risorse nell'account corrente quando utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- OrgID di origine: per filtrare in base all'OrgID di origine, digita tutto o parte dell'ID dell'organizzazione associato alle risorse, come utilizzato in alcune autorizzazioni interservizi in. AWS Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- Origine OrgPaths: per filtrare in base all'origine OrgPaths, digita tutta o parte dell'unità organizzativa (OU) associata alle risorse, come utilizzato in alcune autorizzazioni interservizi in. AWS Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- ID utente: per filtrare in base all'ID utente, digita tutto o parte dell'ID utente IAM di un utente esterno a Account AWS cui è consentito l'accesso alle risorse dell'account corrente. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.
- ID chiave KMS: per filtrare in base all'ID chiave KMS, digita tutto o parte dell'ID chiave per la chiave KMS specificata come condizione per l'accesso crittografato agli oggetti AWS KMS Amazon S3 nel tuo account corrente.
- Destinatari Google: per filtrare in base ai destinatari Google, digita l'ID parziale o completo dell'applicazione Google specificato come condizione per l'accesso del ruolo IAM nell'account corrente. Per ulteriori informazioni, consulta IAM e condition context keys. AWS STS
- Destinatari Cognito: per filtrare in base ai destinatari Amazon Cognito, digita l'ID parziale o completo del pool di identità Amazon Cognito specificato come condizione per l'accesso del ruolo IAM nell'account corrente. Per saperne di più, consulta IAM e AWS STS condition context keys.
- Account chiamante: l' Account AWS ID dell'account che possiede o contiene l'entità chiamante, ad esempio un ruolo IAM, un utente o un utente root dell'account. Viene utilizzato dai servizi che chiamano AWS KMS. Per filtrare in base all'account chiamante, digita l'ID Account AWS parziale o completo.
- ID app Facebook: per filtrare in base all'ID app Facebook, digita l'ID parziale o completo dell'applicazione Facebook (o l'ID del sito) specificato come condizione per consentire l'accesso

con la federazione Facebook a un ruolo IAM nell'account corrente. Per ulteriori informazioni, consulta la sezione id in Chiavi di contesto delle condizioni IAM e AWS STS.

- ID app Amazon: per filtrare in base all'ID app Amazon, digita l'ID parziale o completo dell'applicazione Amazon (o l'ID del sito) specificato come condizione per consentire l'accesso della federazione Login with Amazon a un ruolo IAM nell'account corrente. Per ulteriori informazioni, consulta la sezione id in Chiavi di contesto delle condizioni IAM e AWS STS.
- Token di origine evento Lambda: per filtrare il token di origine evento Lambda passato con le integrazioni Alexa, digita la stringa parziale o completa del token.

Filtraggio di risultati degli accessi inutilizzati

Per filtrare i risultati degli accessi inutilizzati

1. Scegli Accesso inutilizzato, quindi seleziona l'analizzatore dal menu a discesa Visualizza analizzatore.
2. Scegli la casella di ricerca per visualizzare un elenco di proprietà disponibili.
3. Scegliere la proprietà da utilizzare per filtrare i risultati visualizzati.
4. Scegliere il valore da corrispondere per la proprietà. Vengono visualizzati solo i risultati con tale valore presente nel risultato.

   Ad esempio, scegliete Tipo di risultati come proprietà, quindi scegliete Findings type =, quindi scegliete Findings type = unuseDiamRole. Vengono visualizzati solo i risultati con un tipo di unuseDiamRole.

Puoi aggiungere altre proprietà per filtrare ulteriormente i risultati visualizzati. Quando aggiungi altre proprietà, vengono visualizzati solo i risultati che corrispondono a tutte le condizioni del filtro. La definizione di un filtro per visualizzare i risultati che corrispondono a una proprietà O a un'altra proprietà non è supportata. Scegli Cancella filtri per cancellare tutti i filtri definiti e visualizzare tutti i risultati con lo stato specificato per l'analizzatore.

I seguenti campi vengono visualizzati solo quando si visualizzano i risultati di un analizzatore che monitora gli accessi non utilizzati:

- Tipo di risultati: per filtrare in base al tipo di risultato, filtra per tipo di risultati e quindi scegli il tipo di risultato.
- Risorsa: per filtrare in base alla risorsa, digita il nome parziale o completo della risorsa.

- Tipo di risorsa: per filtrare in base al tipo di risorsa, scegli il tipo dall'elenco visualizzato.
- Account del proprietario della risorsa: utilizza questa proprietà per filtrare in base all'account dell'organizzazione proprietaria della risorsa riportata nel risultato.
- ID di ricerca: per filtrare in base all'ID di ricerca, digita tutto o parte dell'ID di ricerca.

Archiviazione dei risultati

Quando ricevi un risultato per l'accesso a una risorsa che è intenzionale, puoi archiviarlo. Ad esempio, un risultato di accesso esterno per un ruolo IAM utilizzato da più utenti per i flussi di lavoro approvati o un risultato di accesso inutilizzato per una chiave di accesso che potrebbe essere ancora necessaria. Quando si archivia un risultato, questo viene cancellato dall'elenco dei risultati attivi. I risultati archiviati non vengono eliminati. Puoi filtrare la pagina Risultati per visualizzare i risultati archiviati e annullarne l'archiviazione in qualsiasi momento.

Per archiviare i risultati dalla pagina Findings (Risultati)

1. Selezionare la casella di controllo accanto a uno o più risultati da archiviare.
2. Scegli Operazioni, quindi seleziona Archivia.

   Viene visualizzata una richiesta di conferma nella parte superiore dello schermo.

Per archiviare i risultati dalla pagina Dettagli risultati

1. Scegliere Finding ID (ID risultato) del risultato da archiviare.
2. Scegliere Archive (Archivia).

   Viene visualizzata una richiesta di conferma nella parte superiore dello schermo.

Per annullare l'archiviazione dei risultati, ripetere i passaggi precedenti, ma scegliere Unarchive (Annulla archiviazione) anziché Archive (Archivia). Quando si annulla l'archiviazione di un risultato, lo stato diventa Attivo.

Risoluzione dei risultati

Risultati dell'accesso esterno

Per risolvere i risultati degli accessi esterni generati dall'accesso che non intendevi consentire, modifica l'istruzione della policy per rimuovere le autorizzazioni che consentono l'accesso alla risorsa identificata. Ad esempio, per i risultati sui bucket Amazon S3, utilizza la console Amazon S3 per configurare le autorizzazioni sul bucket. Per i ruoli IAM, utilizza la console IAM per modificare la policy di attendibilità per il ruolo IAM elencato. Utilizza la console per le altre risorse supportate per modificare le istruzioni della policy che hanno portato a un risultato generato.

Dopo aver apportato una modifica per risolvere un risultato relativo agli accessi esterni, ad esempio la modifica di una policy applicata a un ruolo IAM, Sistema di analisi degli accessi AWS IAM esegue nuovamente la scansione della risorsa. Se la risorsa non è più condivisa al di fuori della zona di attendibilità, lo stato del risultato viene modificato in Risolto. Il risultato non viene più visualizzato nella lista dei risultati attivi, bensì nella lista dei risultati risolti.

> **Note**
>
> Questo non si applica ai risultati Errore. Se Sistema di analisi degli accessi IAM non è in grado di accedere a una risorsa, genera un esito di errore. Se risolvi il problema che ha impedito a Sistema di analisi degli accessi IAM di accedere alla risorsa, l'esito di errore non diventa un risultato risolto, ma viene rimosso del tutto.

Se le modifiche applicate hanno portato alla condivisione della risorsa al di fuori della zona di attendibilità ma in un modo diverso, ad esempio con un principale differente o per un'autorizzazione diversa, Sistema di analisi degli accessi AWS IAM genera un nuovo risultato attivo.

> **Note**
>
> Dopo la modifica di una policy, perché Sistema di analisi degli accessi AWS IAM possa rianalizzare la risorsa e aggiornare il risultato, potrebbero essere necessari fino a 30 minuti. I risultati risolti vengono eliminati 90 giorni dopo l'ultimo aggiornamento dello stato del risultato.

Risultati degli accessi inutilizzati

Per i risultati di Access Analyzer non utilizzati, IAM Access Analyzer fornisce i passaggi consigliati per risolvere i risultati in base al tipo di risultato.

Dopo aver apportato una modifica per risolvere un risultato di accesso inutilizzato, lo stato del risultato viene modificato in Risolto alla successiva esecuzione dell'analizzatore di accessi inutilizzati. Il risultato non viene più visualizzato nell'elenco dei risultati attivi, ma nell'elenco dei risultati risolti. Se si apporta una modifica che risolve solo parzialmente un risultato di accesso inutilizzato, il risultato esistente viene modificato in Risolto ma viene generato un nuovo risultato. Ad esempio, si rimuovono solo alcune delle autorizzazioni inutilizzate di un risultato, ma non tutte.

Sistema di analisi degli accessi AWS IAM addebita i costi per l'analisi degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Risoluzione dei risultati delle autorizzazioni non utilizzati

Per i risultati delle autorizzazioni non utilizzati, IAM Access Analyzer può consigliare politiche da rimuovere da un utente o ruolo IAM e fornire nuove politiche per sostituire le politiche di autorizzazione esistenti. Le raccomandazioni sulle policy non sono supportate per i seguenti scenari:

- Il risultato delle autorizzazioni non utilizzato si riferisce a un utente IAM che fa parte di un gruppo di utenti.
- Il risultato delle autorizzazioni non utilizzato riguarda un ruolo IAM per IAM Identity Center.
- Il risultato delle autorizzazioni non utilizzate include una politica di autorizzazioni esistente che include l'elemento. `notAction`

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso non utilizzato.
3. Scegli un risultato con il tipo di ricerca delle autorizzazioni non utilizzate.
4. Nella sezione Consigli, se ci sono politiche elencate nella colonna Politica consigliata, scegli Anteprima politica per visualizzare la politica esistente con la politica consigliata per sostituire la politica esistente. Se sono presenti più politiche consigliate, puoi scegliere Politica successiva e Politica precedente per visualizzare ogni politica esistente e consigliata.
5. Scegli Scarica JSON per scaricare un file.zip con i file JSON di tutte le politiche consigliate.

6. Crea e collega le policy consigliate all'utente o al ruolo IAM. Per ulteriori informazioni, consulta Modifica delle autorizzazioni per un utente (console) e Modifica della politica di autorizzazione di un ruolo (console).
7. Rimuovi le politiche elencate nella colonna Politica di autorizzazione esistente dall'utente o dal ruolo IAM. Per ulteriori informazioni, consulta Rimuovere le autorizzazioni da un utente (console) e Modifica della politica di autorizzazione di un ruolo (console).

Risoluzione dei problemi relativi ai ruoli non utilizzati

Per i risultati relativi ai ruoli non utilizzati, IAM Access Analyzer consiglia di eliminare il ruolo IAM inutilizzato.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso non utilizzato.
3. Scegli un risultato con il tipo di ricerca del ruolo Non utilizzato.
4. Nella sezione Consigli, esamina i dettagli del ruolo IAM.
5. Elimina il ruolo IAM. Per ulteriori informazioni, consulta Eliminazione di un ruolo IAM (console).

Risoluzione dei problemi relativi alle chiavi di accesso non utilizzate

Per i risultati delle chiavi di accesso non utilizzate, IAM Access Analyzer consiglia di disattivare o eliminare la chiave di accesso inutilizzata.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Accesso non utilizzato.
3. Scegli un risultato con il tipo di ricerca delle chiavi di accesso non utilizzate.
4. Nella sezione Consigli, esamina i dettagli della chiave di accesso.
5. Disattiva o elimina la chiave di accesso. Per ulteriori informazioni, vedere Gestione delle chiavi di accesso (console).

Risoluzione dei ritrovamenti relativi alle password non utilizzate

Per individuare le password non utilizzate, IAM Access Analyzer consiglia di eliminare la password inutilizzata per l'utente IAM.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

2. Scegli Accesso non utilizzato.
3. Scegli una ricerca con il tipo di ricerca della password non utilizzata.
4. Nella sezione Consigli, esamina i dettagli dell'utente IAM.
5. Elimina la password per l'utente IAM. Per ulteriori informazioni, consulta Creazione, modifica o eliminazione di una password utente IAM (console).

Tipi di risorse di Sistema di analisi degli accessi AWS IAM per gli accessi esterni

Per gli analizzatori di accesso esterni, IAM Access Analyzer analizza le politiche basate sulle risorse applicate alle risorse nella regione in cui è stato abilitato IAM Access Analyzer. AWS Vengono analizzate solo le policy basate sulle risorse. Esamina le informazioni di ogni risorsa per i dettagli su come Sistema di analisi degli accessi AWS IAM genera i risultati per ogni tipo di risorsa.

> **Note**
>
> I tipi di risorse supportati elencati sono per analizzatori degli accessi esterni. Gli analizzatori degli accessi inutilizzati supportano solo utenti e ruoli IAM. Per ulteriori informazioni, consulta Uso dei risultati.

Tipi di risorse supportati per gli accessi esterni:

- Bucket Amazon Simple Storage Service
- Bucket di directory di Amazon Simple Storage Service
- AWS Identity and Access Management ruoli
- AWS Key Management Service chiavi
- AWS Lambda funzioni e livelli
- Code Amazon Simple Queue Service
- AWS Secrets Manager segreti
- Argomenti su Amazon Simple Notification Service
- Volumi e snapshot di Amazon Elastic Block Store
- Amazon Relational Database Service
- Snapshot di cluster di database di Amazon Relational Database Service

- Repository di Amazon Elastic Container Registry
- File system di Amazon Elastic File System
- Flussi Amazon DynamoDB
- Tabelle Amazon DynamoDB

Bucket Amazon Simple Storage Service

Quando Sistema di analisi degli accessi AWS IAM analizza i bucket Amazon S3, genera un risultato quando la policy di un bucket Amazon S3, una ACL o un punto di accesso applicato a un bucket concede l'accesso a un'entità esterna. Un'entità esterna è un'entità principale o un'altra entità che puoi utilizzare per creare un filtro che non si trova all'interno della zona di attendibilità. Ad esempio, se la policy di un bucket concede l'accesso a un altro account o consente l'accesso pubblico, Sistema di analisi degli accessi AWS IAM genera un risultato. Tuttavia, se abiliti Block Public Access (Blocca accesso pubblico) nel bucket, puoi bloccare l'accesso a livello di account o bucket.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM non analizza la policy dei punti di accesso associata ai punti di accesso multi-account perché il punto di accesso e la relativa policy sono esterni all'account dell'analizzatore. Sistema di analisi degli accessi AWS IAM genera un risultato pubblico quando un bucket delega l'accesso a un punto di accesso multi-account e il blocco dell'accesso pubblico non è abilitato sul bucket o sull'account. Quando abiliti l'opzione di blocco dell'accesso pubblico, il rilevamento pubblico viene risolto e Sistema di analisi degli accessi AWS IAM genera un risultato tra account per il punto di accesso multi-account.

Le impostazioni per il blocco dell'accesso pubblico di Amazon S3 sostituiscono le policy applicate al bucket. Le impostazioni sovrascrivono anche le policy applicate ai punti di accesso del bucket. Sistema di analisi degli accessi AWS IAM analizza le impostazioni del blocco dell'accesso pubblico a livello di bucket ogni volta che cambia una policy. Tuttavia, valuta le impostazioni del blocco dell'accesso pubblico a livello di account solo una volta ogni 6 ore. Ciò significa che Sistema di analisi degli accessi AWS IAM potrebbe non generare o risolvere un risultato per l'accesso pubblico a un bucket per un massimo di 6 ore. Ad esempio, se hai una policy del bucket che consente l'accesso pubblico, Sistema di analisi degli accessi AWS IAM genera un risultato per tale accesso. Se quindi abiliti il blocco dell'accesso pubblico per bloccare tutti gli accessi pubblici al bucket a livello di account, Sistema di analisi degli accessi AWS IAM non risolve il risultato per la policy del bucket per

un massimo di 6 ore, anche se tutti gli accessi pubblici al bucket sono bloccati. La risoluzione dei dati pubblici relativi ai punti di accesso multi-account può inoltre richiedere fino a 6 ore dopo l'abilitazione del blocco dell'accesso pubblico a livello di account.

Per un punto di accesso multi-regione, Sistema di analisi degli accessi AWS IAM utilizza una policy stabilita per la generazione dei risultati. Sistema di analisi degli accessi AWS IAM valuta le modifiche apportate ai punti di accesso multi-regione una volta ogni 6 ore. Ciò significa che Sistema di analisi degli accessi AWS IAM non genera né risolve un risultato per un massimo di 6 ore, anche se viene creato o eliminato un punto di accesso multi-regione o se ne aggiorna la policy.

Bucket di directory di Amazon Simple Storage Service

I bucket di directory di Amazon S3 utilizzano la classe di storage Amazon S3 Express One, consigliata per carichi di lavoro o applicazioni critici in termini di prestazioni. Per i bucket di directory di Amazon S3, Sistema di analisi degli accessi AWS IAM analizza la relativa policy, incluse le istruzioni sulle condizione in una policy, che consentono a un'entità esterna di accedere a un bucket di directory. Per ulteriori informazioni sui bucket di directory di Amazon S3, consulta Bucket di directory nella Guida per l'utente di Amazon Simple Storage Service.

AWS Identity and Access Management ruoli

Per i ruoli IAM, Sistema di analisi degli accessi AWS IAM analizza le policy di attendibilità. In una policy di attendibilità per il ruolo si definiscono le entità attendibili per assumere il ruolo. Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo in IAM. Sistema di analisi degli accessi AWS IAM genera i risultati per i ruoli all'interno della zona di attendibilità a cui può accedere un'entità esterna che si trova al di fuori della zona di attendibilità.

> **Note**
>
> Un ruolo IAM è una risorsa globale. Se una policy di attendibilità per il ruolo concede l'accesso a un'entità esterna, Sistema di analisi degli accessi AWS IAM genera un risultato in ogni regione abilitata.

AWS Key Management Service chiavi

Infatti AWS KMS keys, IAM Access Analyzer analizza le politiche e le concessioni chiave applicate a una chiave. Sistema di analisi degli accessi AWS IAM genera un risultato se una policy di chiave

o una concessione consente a un'entità esterna di accedere alla chiave. Ad esempio, se utilizzi la chiave kms: CallerAccount condition in un'informativa politica per consentire l'accesso a tutti gli utenti di un AWS account specifico e specifichi un account diverso dall'account corrente (la zona di fiducia per l'analizzatore corrente), IAM Access Analyzer genera un risultato. Per ulteriori informazioni sulle chiavi di AWS KMS condizione nelle dichiarazioni delle politiche IAM, consulta AWS KMS Condition Keys.

Quando Sistema di analisi degli accessi AWS IAM analizza una chiave KMS, legge i metadati della chiave, ad esempio la policy della chiave e l'elenco delle concessioni. Se la policy della chiave non consente al ruolo di Sistema di analisi degli accessi AWS IAM di leggere i metadati della chiave, viene generato un errore di accesso negato. Ad esempio, se l'istruzione della policy di esempio seguente è l'unica policy applicata a una chiave, viene generato un errore di accesso negato in Sistema di analisi degli accessi AWS IAM:

```
{
    "Sid": "Allow access for Key Administrators",
    "Effect": "Allow",
    "Principal": {
       "AWS": "arn:aws:iam::111122223333:role/Admin"
    },
    "Action": "kms:*",
    "Resource": "*"
}
```

Poiché questa istruzione consente solo al ruolo denominato Admin dell' AWS account 111122223333 di accedere alla chiave, viene generata una ricerca di errore di accesso negato perché IAM Access Analyzer non è in grado di analizzare completamente la chiave. Un risultato di errore viene visualizzato in rosso nella tabella Findings (Risultati). Il risultato è simile al seguente:

```
{
    "error": "ACCESS_DENIED",
    "id": "12345678-1234-abcd-dcba-111122223333",
    "analyzedAt": "2019-09-16T14:24:33.352Z",
    "resource": "arn:aws:kms:us-
west-2:1234567890:key/1a2b3c4d-5e6f-7a8b-9c0d-1a2b3c4d5e6f7g8a",
    "resourceType": "AWS::KMS::Key",
    "status": "ACTIVE",
    "updatedAt": "2019-09-16T14:24:33.352Z"
}
```

Quando crei una chiave KMS, le autorizzazioni concesse per accedere alla chiave dipendono dalla modalità di creazione della chiave. Se viene visualizzato un errore di tipo Accesso negato per una risorsa chiave, applica la seguente istruzione della policy alla risorsa chiave per concedere a Sistema di analisi degli accessi AWS IAM l'autorizzazione per accedere alla chiave.

```
{
    "Sid": "Allow IAM Access Analyzer access to key metadata",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/aws-service-role/access-
analyzer.amazonaws.com/AWSServiceRoleForAccessAnalyzer"
        },
    "Action": [
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:List*"
    ],
    "Resource": "*"
},
```

Dopo aver ricevuto un risultato di accesso negato per una risorsa chiave KMS e quindi averlo risolto aggiornando la policy della chiave, il risultato viene aggiornato sullo stato Risolto. Se sono presenti istruzioni di policy o concessioni della chiave che concedono l'autorizzazione alla chiave per un'entità esterna, è possibile che vengano visualizzati ulteriori risultati per la risorsa chiave.

AWS Lambda funzioni e livelli

Per quanto riguarda AWS Lambda le funzioni, IAM Access Analyzer analizza le policy, incluse le dichiarazioni di condizione contenute in una policy, che concedono l'accesso alla funzione a un'entità esterna. Con Lambda, puoi associare policy uniche basate sulle risorse a funzioni, versioni, alias e layer. IAM Access Analyzer riporta gli accessi esterni in base a policy basate sulle risorse collegate a funzioni e livelli. IAM Access Analyzer non segnala l'accesso esterno in base a policy basate su risorse collegate ad alias e versioni specifiche richiamate utilizzando un ARN qualificato.

Per ulteriori informazioni, consulta Using Resource-based policy for Lambda e Using versions nella Developer Guide. AWS Lambda

Code Amazon Simple Queue Service

Per le code Amazon SQS, Sistema di analisi degli accessi AWS IAM analizza le policy, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a una coda.

AWS Secrets Manager segreti

Per quanto riguarda AWS Secrets Manager i segreti, IAM Access Analyzer analizza le policy, incluse le condizioni contenute in una policy, che consentono a un'entità esterna di accedere a un segreto.

Argomenti su Amazon Simple Notification Service

Sistema di analisi degli accessi AWS IAM analizza le policy basate sulle risorse allegate agli argomenti di Amazon SNS, incluse le istruzioni delle condizioni nelle policy che consentono l'accesso esterno a un argomento. Puoi consentire agli account esterni di eseguire azioni Amazon SNS come la sottoscrizione e la pubblicazione di argomenti tramite una policy basata sulle risorse. Un argomento Amazon SNS è accessibile dall'esterno se i principali di un account esterno alla tua zona di fiducia possono eseguire operazioni sull'argomento. Se scegli `Everyone` nella tua policy quando crei un argomento Amazon SNS, rendi l'argomento accessibile al pubblico. `AddPermission` è un altro modo per aggiungere una policy basata sulle risorse a un argomento Amazon SNS che consente l'accesso esterno.

Volumi e snapshot di Amazon Elastic Block Store

Gli snapshot di volumi di Amazon Elastic Block Store non hanno policy basate sulle risorse. Uno snapshot viene condiviso tramite le autorizzazioni di condivisione di Amazon EBS. Per gli snapshot di volume Amazon EBS, Sistema di analisi degli accessi AWS IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a uno snapshot. Se crittografato, uno snapshot di volume Amazon EBS può essere condiviso con account esterni. Uno snapshot di volume non crittografato può essere condiviso con account esterni e garantire l'accesso pubblico. Le impostazioni di condivisione sono nell'attributo `CreateVolumePermissions` dello snapshot. Quando i clienti visualizzano in anteprima l'accesso esterno di uno snapshot di Amazon EBS, possono specificare la chiave di crittografia come indicatore del fatto che lo snapshot è crittografato, in modo simile a come l'anteprima di Sistema di analisi degli accessi AWS IAM gestisce i segreti di Gestione dei segreti.

Amazon Relational Database Service

Gli snapshot del database Amazon RDS non hanno policy basate su risorse. Uno snapshot del database viene condiviso tramite le autorizzazioni del database Amazon RDS e possono essere condivisi solo snapshot manuali del database. Per gli snapshot del database Amazon RDS, Sistema di analisi degli accessi AWS IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a uno snapshot. Gli snapshot del database non crittografati possono essere

pubblici. Gli snapshot del database crittografati non possono essere condivisi pubblicamente, ma possono essere condivisi con un massimo di altri 20 account. Per ulteriori informazioni, consulta Creazione di uno snapshot DB. Sistema di analisi degli accessi AWS IAM considera la capacità di esportare uno snapshot manuale del database (ad esempio, in un bucket Amazon S3) come accesso attendibile.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM non identifica l'accesso pubblico o l'accesso multi-account configurato direttamente sul database stesso. Sistema di analisi degli accessi AWS IAM identifica solo i risultati per l'accesso pubblico o l'accesso multi-account configurati nello snapshot del database di Amazon RDS.

Snapshot di cluster di database di Amazon Relational Database Service

Gli snapshot del cluster di database Amazon RDS non hanno policy basate su risorse. Uno snapshot viene condiviso tramite le autorizzazioni del cluster di database di Amazon RDS. Per gli snapshot del cluster di database di Amazon RDS, Sistema di analisi degli accessi AWS IAM analizza le liste di controllo degli accessi che consentono a un'entità esterna di accedere a uno snapshot. Gli snapshot del cluster non crittografati possono essere pubblici. Gli snapshot del cluster crittografati non possono essere condivisi pubblicamente. Gli snapshot del cluster non crittografati e crittografati possono essere condivisi con al massimo altri 20 account. Per ulteriori informazioni, consulta la sezione Creating a DB Cluster Snapshot (Creazione di uno snapshot cluster database). Sistema di analisi degli accessi AWS IAM considera la capacità di esportare uno snapshot del cluster di database (ad esempio, in un bucket Amazon S3) come accesso attendibile.

> **Note**
>
> I risultati di IAM Access Analyzer non includono il monitoraggio di alcuna condivisione di cluster e cloni di Amazon RDS DB con altri Account AWS utenti o organizzazioni. AWS Resource Access Manager Sistema di analisi degli accessi AWS IAM identifica solo i risultati per l'accesso pubblico o l'accesso multi-account configurati nello snapshot del cluster di database di Amazon RDS.

Repository di Amazon Elastic Container Registry

Per i repository Amazon ECR, Sistema di analisi degli accessi AWS IAM analizza le policy basate su risorse, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a un repository (in modo simile ad altri tipi di risorse come gli argomenti di Amazon SNS e i file system Amazon EFS). Per i repository Amazon ECR, un principale deve avere l'autorizzazione per `ecr:GetAuthorizationToken` tramite una policy basata sull'identità per essere considerato disponibile esternamente.

File system di Amazon Elastic File System

Per le code Amazon SQS, Sistema di analisi degli accessi AWS IAM analizza le policy, incluse le istruzioni di condizione in una policy che consentono a un'entità esterna di accedere a una coda. Un file system Amazon EFS è accessibile dall'esterno se i principali di un account esterno alla tua zona di attendibilità possono eseguire operazioni su quel file system. L'accesso è definito da una policy del file system che utilizza IAM e da come viene montato il file system. Ad esempio, il montaggio del file system Amazon EFS in un altro account è considerato accessibile dall'esterno, a meno che tale account non si trovi nella tua organizzazione e tu non abbia definito l'organizzazione come la tua zona di attendibilità. Se monti il file system da un cloud privato virtuale con una sottorete pubblica, il file system sarà accessibile dall'esterno. Quando usi Amazon EFS con AWS Transfer Family, le richieste di accesso al file system ricevute da un server Transfer Family di proprietà di un account diverso da quello del file system vengono bloccate se il file system consente l'accesso pubblico.

Flussi Amazon DynamoDB

IAM Access Analyzer genera un risultato se una policy DynamoDB consente almeno un'azione tra account che consente a un'entità esterna di accedere a un flusso DynamoDB. Per ulteriori informazioni sulle azioni multiaccount supportate per DynamoDB, consulta le azioni IAM supportate da politiche basate sulle risorse nella Amazon DynamoDB Developer Guide.

Tabelle Amazon DynamoDB

IAM Access Analyzer genera un risultato per una tabella DynamoDB se una policy DynamoDB consente almeno un'azione tra account che consente a un'entità esterna di accedere a una tabella o a un indice DynamoDB. Per ulteriori informazioni sulle azioni multiaccount supportate per DynamoDB, consulta le azioni IAM supportate da politiche basate sulle risorse nella Amazon DynamoDB Developer Guide.

Impostazioni per Sistema di analisi degli accessi AWS IAM

Se esegui la configurazione AWS Identity and Access Management Access Analyzer nel tuo account di AWS Organizations gestione, puoi aggiungere un account membro nell'organizzazione come amministratore delegato per gestire IAM Access Analyzer per la tua organizzazione. L'amministratore delegato dispone delle autorizzazioni per creare e gestire gli analizzatori nell'organizzazione. Solo l'account di gestione può aggiungere un amministratore delegato.

Amministratore delegato per Sistema di analisi degli accessi AWS IAM.

L'amministratore delegato per Sistema di analisi degli accessi AWS IAM è un account membro all'interno dell'organizzazione che dispone delle autorizzazioni per creare e gestire gli analizzatori che analizzano gli accessi nell'organizzazione. Solo l'account di gestione può aggiungere, rimuovere o modificare un amministratore delegato.

Se aggiungi un amministratore delegato, puoi in un secondo momento passare a un account diverso per l'amministratore delegato. In questo caso, l'account amministratore delegato precedente perde l'autorizzazione per tutti gli analizzatori creati utilizzando tale account per analizzare gli accessi nell'organizzazione. Questi analizzatori passano a uno stato disabilitato e non generano più nuovi risultati, né aggiornano quelli esistenti. Anche i risultati esistenti per questi analizzatori non sono più accessibili. Puoi accedervi nuovamente in futuro configurando l'account come amministratore delegato. Se ritieni che uno stesso account di amministratore delegato non verrà più utilizzato, è consigliabile eliminare gli analizzatori prima di modificare l'amministratore delegato. Questa operazione elimina tutti i risultati generati. Quando il nuovo amministratore delegato crea nuovi analizzatori, vengono generate nuove istanze degli stessi risultati. Non perdi alcun risultato, vengono solo generati per il nuovo analizzatore in un altro account. Puoi continuare ad accedere ai risultati dell'organizzazione anche utilizzando l'account di gestione dell'organizzazione che dispone anche di autorizzazioni di amministratore. Il nuovo amministratore delegato deve creare i nuovi analizzatori per Sistema di analisi degli accessi AWS IAM per avviare il monitoraggio delle risorse nell'organizzazione.

Se l'amministratore delegato lascia l' AWS organizzazione, i privilegi di amministrazione delegata vengono rimossi dall'account. Tutti gli analizzatori nell'account con l'organizzazione come zona di attendibilità passano a uno stato disabilitato. Anche i risultati esistenti per questi analizzatori non sono più accessibili.

La prima volta che si configurano gli analizzatori nell'account di gestione, puoi scegliere l'opzione Aggiungi amministratore delegato nella pagina Impostazioni dell'analizzatore nella console di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM addebita i costi per gli analizzatori degli accessi inutilizzati in base al numero di ruoli e utenti IAM analizzati ogni mese da ogni analizzatore. Se crei un analizzatore degli accessi inutilizzati nell'account di gestione e uno nell'account dell'amministratore delegato, ti verranno addebitati i costi per entrambi gli analizzatori. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Per aggiungere un amministratore delegato utilizzando la console

1. Accedi alla AWS console utilizzando l'account di gestione dell'organizzazione.
2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. In Access Analyzer, scegli Impostazioni dell'analizzatore.
4. Scegliere Add delegated administrator (Aggiungi amministratore delegato).
5. Nel campo Amministratore delegato, inserisci il numero Account AWS di un account membro dell'organizzazione per creare l'amministratore delegato.

   L'account deve essere un membro dell'organizzazione.
6. Seleziona Salvataggio delle modifiche.

Per aggiungere un amministratore delegato utilizzando AWS CLI o gli SDK AWS

Quando crei un analizzatore per analizzare l'accesso all'intera organizzazione in un account amministratore delegato utilizzando la AWS CLI, l' AWS API (utilizzando gli AWS SDK) oppure AWS CloudFormation, devi utilizzare AWS Organizations le API per abilitare l'accesso al servizio per IAM Access Analyzer e registrare l'account membro come amministratore delegato.

1. Abilita l'accesso affidabile ai servizi per IAM Access Analyzer in. AWS Organizations Scopri come abilitare o disabilitare l'accesso affidabile nella Guida per l' AWS Organizations utente.
2. Registra un account membro valido della tua AWS organizzazione come amministratore delegato utilizzando l'operazione AWS Organizations `RegisterDelegatedAdministrator`API o il `register-delegated-administrator` AWS CLI comando.

Dopo aver cambiato l'amministratore delegato, il nuovo amministratore deve creare gli analizzatori per avviare il monitoraggio dell'accesso alle risorse dell'organizzazione.

Eliminazione degli analizzatori

È possibile eliminare gli analizzatori degli accessi esterni e inutilizzati esistenti dalla pagina Impostazioni dell'analizzatore. Quando si elimina un analizzatore, le risorse specificate nell'analizzatore non vengono più monitorate e non vengono generati nuovi risultati. Tutti i risultati generati dall'analizzatore vengono eliminati.

Per i risultati che vengono eliminati perché l'analizzatore che li ha generati viene eliminato, l'evento viene inviato a EventBridge nei due giorni successivi all'eliminazione dell'analizzatore. Dopo l'eliminazione dell'analizzatore, possono essere necessari fino a 90 giorni prima che i risultati di Security Hub vengano eliminati.

Per eliminare un analizzatore

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Access Analyzer, scegli Impostazioni dell'analizzatore.
3. Seleziona l'analizzatore da eliminare, quindi scegli Elimina.
4. Nella casella di conferma, scrivi **`delete`** e quindi scegli Elimina.

Regole di archiviazione

Le regole di archiviazione archiviano automaticamente i nuovi risultati che soddisfano i criteri che definisci quando crei la regola. Puoi inoltre applicare le regole di archiviazione retroattivamente per archiviare i risultati esistenti che soddisfano i criteri delle regole di archiviazione. Ad esempio, puoi creare una regola di archiviazione per archiviare automaticamente tutti i risultati per un bucket Amazon S3 specifico a cui concedi regolarmente l'accesso. In alternativa, se concedi a un'entità specifica l'accesso a più risorse, è possibile creare una regola che archivia automaticamente qualsiasi nuovo risultato generato per l'accesso concesso a tale entità. In questo modo è possibile concentrarsi solo sui risultati attivi che possono indicare un rischio per la sicurezza.

Quando crei una regola di archiviazione, solo i nuovi risultati che corrispondono ai criteri della regola vengono archiviati automaticamente. I risultati esistenti non vengono archiviati automaticamente. Quando crei una regola, puoi includere fino a 20 valori per criterio. Per l'elenco delle chiavi di filtro che puoi utilizzare per creare o aggiornare una regola di archivio, consulta Chiavi di filtro di Sistema di analisi degli accessi AWS IAM.

> **Note**
>
> Quando crei o modifichi una regola di archiviazione, Sistema di analisi degli accessi AWS IAM non convalida i valori inclusi nel filtro per la regola. Ad esempio, se aggiungi una regola per la corrispondenza con un Account AWS, Sistema di analisi degli accessi AWS IAM accetta qualsiasi valore nel campo, anche se non è un numero di account AWS valido.

Per creare una regola di archiviazione

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Access Analyzer e poi Impostazioni dell'analizzatore.
3. Nella sezione Analizzatori, scegli l'analizzatore per il quale desideri creare una regola di archivio.
4. Nella scheda Regole di archivio, scegli Crea regola di archivio.
5. Immettere un nome per la regola se si desidera modificare il nome predefinito.
6. Nella sezione Rule (Regola) in Criteria (Criteri), selezionare una proprietà da corrispondere per la regola.
7. Scegli una condizione per il valore della proprietà, ad esempio Contiene, È o Non è uguale.

   Gli operatori disponibili dipendono dalla proprietà scelta.
8. Facoltativamente, aggiungere altri valori per la proprietà o altri criteri per la regola. Per i risultati degli accessi esterni, per assicurarti che la regola non archivi i nuovi risultati per l'accesso pubblico, puoi anche includere il criterio Accesso pubblico e impostarlo su false.

   Per aggiungere un altro valore per un criterio, scegliere Add another value (Aggiungi un altro valore). Per aggiungere un altro criterio per la regola, scegli Aggiungi criterio.
9. Al termine dell'aggiunta di criteri e valori, scegli Create rule (Crea regola) per applicare la regola solo ai nuovi risultati. Scegli Create and archive active findings (Crea e archivi i risultati attivi) per archiviare i risultati nuovi ed esistenti in base ai criteri della regola. Nella sezione Results (Risultati) puoi esaminare l'elenco dei risultati attivi a cui si applica la regola di archiviazione.

Ad esempio, per creare una regola per i risultati degli accessi esterni che archivia automaticamente tutti i risultati per i bucket Amazon S3: scegliere Tipo di risorsa e poi è per la condizione. Quindi scegli Bucket S3 dall'elenco Valori.

Per creare una regola per i risultati degli accessi inutilizzati che archivia automaticamente tutti i risultati per un determinato account, scegli Account del proprietario della risorsa e poi Uguale per la condizione. Digitate l' Account AWS ID nella casella di testo Valore.

Continua a definire i criteri per personalizzare la regola in base all'ambiente, quindi scegli Crea regola.

Se si crea una nuova regola e si aggiungono più criteri, è possibile rimuovere un singolo criterio dalla regola scegliendo Remove this criterion (Rimuovi questo criterio). È possibile rimuovere un valore aggiunto per un criterio scegliendo Remove value (Rimuovi valore).

Per modificare una regola di archiviazione

1. Scegli il nome della regola da modificare nella colonna Nome.

   È possibile modificare una sola regola di archiviazione alla volta.
2. Per ogni criterio, aggiungi nuovi criteri e valori o rimuovi quelli esistenti.
3. Scegli Save changes (Salva modifiche) per applicare la regola solo ai nuovi risultati. Scegli Save and archive active findings (Salva e archivia i risultati attivi) per archiviare i risultati nuovi ed esistenti in base ai criteri della regola.

Per eliminare una regola di archiviazione

1. Seleziona la casella di controllo per la regola da eliminare.
2. Scegli Elimina.
3. Digitare **`delete`** nella finestra di dialogo di conferma Delete archive rule (Elimina regola di archiviazione) e quindi scegliere Delete (Elimina).

Le regole vengono eliminate solo dall'analizzatore nella regione corrente. È necessario eliminare le regole di archiviazione separatamente per ogni analizzatore creato in altre regioni.

Monitoraggio AWS Identity and Access Management Access Analyzer con Amazon EventBridge

Utilizza le informazioni contenute in questo argomento per scoprire come monitorare i risultati di IAM Access Analyzer e accedere alle anteprime con Amazon. EventBridge EventBridge è la nuova versione di Amazon CloudWatch Events.

Eventi dei risultati

IAM Access Analyzer invia un evento EventBridge per ogni risultato generato, per modificare lo stato di un risultato esistente e quando un risultato viene eliminato. Per ricevere risultati e notifiche sui risultati, devi creare una regola di evento in Amazon EventBridge. Quando si crea una regola di evento, è anche possibile specificare un'operazione di destinazione da attivare in base alla regola. Ad esempio, puoi creare una regola di evento che attiva un argomento Amazon SNS quando viene ricevuto un evento per un nuovo risultato da Sistema di analisi degli accessi AWS IAM.

Accesso a eventi di anteprima

IAM Access Analyzer invia un evento EventBridge a ogni anteprima di accesso e modifica del relativo stato. Ciò include un evento quando viene creata per la prima volta l'anteprima di accesso (stato Creazione), quando l'anteprima di accesso è completata (stato Completato) o quando la creazione dell'anteprima di accesso non è riuscita (stato Non riuscito). Per ricevere notifiche sulle anteprime di accesso, devi creare una regola di evento in. EventBridge Quando crei una regola di evento, puoi anche specificare un'operazione di destinazione da attivare in base alla regola. Ad esempio, puoi creare una regola di evento che attiva un argomento Amazon SNS quando viene ricevuta un'anteprima dell'accesso completato da Sistema di analisi degli accessi AWS IAM.

Frequenza delle notifiche di evento

IAM Access Analyzer invia gli eventi relativi a nuove scoperte e scoperte con aggiornamenti di stato EventBridge entro circa un'ora dal momento in cui si verifica l'evento nel tuo account. IAM Access Analyzer invia anche eventi EventBridge quando un risultato risolto viene eliminato perché il periodo di conservazione è scaduto. Per i risultati che vengono eliminati perché l'analizzatore che li ha generati viene eliminato, l'evento viene inviato a EventBridge circa 24 ore dall'eliminazione dell'analizzatore. Quando un risultato viene eliminato, lo stato del risultato non viene modificato. L'attributo `isDeleted` viene impostato su `true`. IAM Access Analyzer invia anche eventi per le anteprime di accesso appena create e le modifiche allo stato delle anteprime di accesso a. EventBridge

Esempi di eventi relativi ai risultati degli accessi esterni

Di seguito è riportato un esempio di evento di ricerca degli accessi esterni di IAM Access Analyzer inviato a. EventBridge L'`id`elenco è l'ID dell'evento in EventBridge. Per ulteriori informazioni, consulta Eventi e modelli di eventi in EventBridge.

Nell'oggetto `detail`, i valori per gli attributi `accountId` e `region` si riferiscono all'account e alla regione riportati nel risultato. L'attributo `isDeleted` indica se l'evento è derivato dal risultato eliminato. L'`id` è l'ID risultato. L'array `resources` è un singleton con l'ARN dell'analizzatore che ha generato il risultato.

```
{
    "account": "111122223333",
    "detail": {
        "accountId": "111122223333",
        "action": [
            "s3:GetObject"
        ],
        "analyzedAt": "2019-11-21T01:22:22Z",
        "condition": {},
        "createdAt": "2019-11-20T04:58:50Z",
        "id": "22222222-dcba-4444-dcba-333333333333",
        "isDeleted": false,
        "isPublic": false,
        "principal": {
            "AWS": "999988887777"
        },
        "region": "us-west-2",
        "resource": "arn:aws:s3:::my-bucket",
        "resourceType": "AWS::S3::Bucket",
        "status": "ACTIVE",
        "updatedAt": "2019-11-21T01:14:07Z",
        "version": "1.0"
    },
    "detail-type": "Access Analyzer Finding",
    "id": "11111111-2222-4444-aaaa-333333333333",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2019-11-21T01:22:33Z",
    "version": "0"
}
```

IAM Access Analyzer invia anche gli eventi a EventBridge per rilevare gli errori. Un risultato di errore è generato quando Sistema di analisi degli accessi AWS IAM non è in grado di analizzare la risorsa. Gli eventi per i risultati di errore includono un attributo `error` come illustrato nell'esempio seguente.

```
{
    "account": "111122223333",
    "detail": {
        "accountId": "111122223333",
        "analyzedAt": "2019-11-21T01:22:22Z",
        "createdAt": "2019-11-20T04:58:50Z",
        "error": "ACCESS_DENIED",
        "id": "22222222-dcba-4444-dcba-333333333333",
        "isDeleted": false,
        "region": "us-west-2",
        "resource": "arn:aws:s3:::my-bucket",
        "resourceType": "AWS::S3::Bucket",
        "status": "ACTIVE",
        "updatedAt": "2019-11-21T01:14:07Z",
        "version": "1.0"
    },
    "detail-type": "Access Analyzer Finding",
    "id": "11111111-2222-4444-aaaa-333333333333",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2019-11-21T01:22:33Z",
    "version": "0"
}
```

Esempi di eventi relativi ai risultati degli accesso inutilizzati

Di seguito è riportato un esempio di evento di ricerca degli accessi non utilizzati di IAM Access Analyzer inviato a. EventBridge L'`id`elenco è l'ID dell'evento in. EventBridge Per ulteriori informazioni, consulta Eventi e modelli di eventi in EventBridge.

Nell'oggetto `detail`, i valori per gli attributi `accountId` e `region` si riferiscono all'account e alla regione riportati nel risultato. L'attributo `isDeleted` indica se l'evento è derivato dal risultato eliminato. L'`id` è l'ID risultato.

```
{
    "version": "0",
    "id": "dc7ce3ee-114b-3243-e249-7f10f9054b21",
    "detail-type": "Unused Access Finding for IAM entities",
    "source": "aws.access-analyzer",
```

```
    "account": "123456789012",
    "time": "2023-09-29T17:31:40Z",
    "region": "us-west-2",
    "resources": [
       "arn:aws:access-analyzer:us-west-2:123456789012:analyzer/
integTestLongLivingAnalyzer-DO-NOT-DELETE"
       ],
    "detail": {
        "findingId": "b8ae0460-5d29-4922-b92a-ba956c986277",
        "resource": "arn:aws:iam::111122223333:role/FindingIntegTestFakeRole",
        "resourceType": "AWS::IAM::Role",
        "accountId": "111122223333",
        "createdAt": "2023-09-29T17:29:18.758Z",
        "updatedAt": "2023-09-29T17:29:18.758Z",
        "analyzedAt": "2023-09-29T17:29:18.758Z",
        "previousStatus": "",
        "status": "ACTIVE",
        "version": "62160bda-8e94-46d6-ac97-9670930d8ffb",
        "isDeleted": false,
        "findingType": "UnusedPermission",
        "numberOfUnusedServices": 0,
        "numberOfUnusedActions": 1
        }
    }
```

IAM Access Analyzer invia anche gli eventi a EventBridge per rilevare gli errori. Un risultato di errore è generato quando Sistema di analisi degli accessi AWS IAM non è in grado di analizzare la risorsa. Gli eventi per i risultati di errore includono un attributo `error` come illustrato nell'esempio seguente.

```
{
    "version": "0",
    "id": "c2e7aa1a-4df7-7652-f33e-64113b8997d4",
    "detail-type": "Unused Access Finding for IAM entities",
    "source": "aws.access-analyzer",
    "account": "111122223333",
    "time": "2023-10-31T20:26:12Z",
    "region": "us-west-2",
    "resources": [
      "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ba811f91-
de99-41a4-97c0-7481898b53f2"
      ],
    "detail": {
        "findingId": "b01a34f2-e118-46c9-aef8-0d8526b495c7",
```

```
        "resource": "arn:aws:iam::123456789012:role/TestRole",
        "resourceType": "AWS::IAM::Role",
        "accountId": "444455556666",
        "createdAt": "2023-10-31T20:26:08.647Z",
        "updatedAt": "2023-10-31T20:26:09.245Z",
        "analyzedAt": "2023-10-31T20:26:08.525Z",
        "previousStatus": "",
        "status": "ACTIVE",
        "version": "7c7a72a2-7963-4c59-ac71-f0be597010f7",
        "isDeleted": false,
        "findingType": "UnusedIAMRole",
        "error": "INTERNAL_ERROR"
        }
  }
```

Esempio di eventi di anteprima di accesso

L'esempio seguente mostra i dati per il primo evento a cui viene inviato EventBridge quando si crea un'anteprima di accesso. L'array `resources` è un singleton con l'ARN dell'analizzatore a cui è associata l'anteprima di accesso. Nell'oggetto `detail`, `id` si riferisce all'ID di anteprima dell'accesso e `configuredResources` fa riferimento alla risorsa per la quale è stata creata l'anteprima di accesso. `status` è `Creating` e fa riferimento allo stato dell'anteprima dell'accesso. `previousStatus` non è specificato perché l'anteprima di accesso è stata appena creata.

```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::example-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.00Z",
        "region": "us-west-2",
        "status": "CREATING",
        "version": "1.0"
    },
    "detail-type": "Access Preview State Change",
    "id": "aaaabbbb-2222-3333-4444-555566667777",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
```

```
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```

L'esempio seguente mostra i dati di un evento inviato a EventBridge per un'anteprima di accesso con una modifica dello stato da `Creating` a`Completed`. Nell'oggetto dettaglio, `id` fa riferimento all'ID di anteprima dell'accesso. `status` e `previousStatus` fanno riferimento allo stato dell'anteprima dell'accesso, in cui lo stato precedente era `Creating` e lo stato corrente è `Completed`.

```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::example-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.000Z",
        "previousStatus": "CREATING",
        "region": "us-west-2",
        "status": "COMPLETED",
        "version": "1.0"
    },
    "detail-type": "Access Preview State Change",
    "id": "11112222-3333-4444-5555-666677778888",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```

L'esempio seguente mostra i dati di un evento inviato a EventBridge per un'anteprima di accesso con una modifica dello stato da `Creating` a`Failed`. Nell'oggetto `detail`, `id` fa riferimento all'ID di anteprima dell'accesso. `status` e `previousStatus` fanno riferimento allo stato dell'anteprima dell'accesso, in cui lo stato precedente era `Creating` e lo stato corrente è `Failed`. Il campo `statusReason` fornisce il codice motivo che indica che l'anteprima di accesso non è riuscita a causa di una configurazione di risorse non valida.

```
{
    "account": "111122223333",
    "detail": {
        "accessPreviewId": "aaaabbbb-cccc-dddd-eeee-ffffaaaabbbb",
        "configuredResources": [
            "arn:aws:s3:::example-bucket"
        ],
        "createdAt": "2020-02-20T00:00:00.00Z",
        "previousStatus": "CREATING",
        "region": "us-west-2",
        "status": "FAILED",
        "statusReason": {
            "code": "INVALID_CONFIGURATION"
        },
        "version": "1.0"
    },
    "detail-type": "Access Preview State Change",
    "id": "99998888-7777-6666-5555-444433332222",
    "region": "us-west-2",
    "resources": [
        "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/MyAnalyzer"
    ],
    "source": "aws.access-analyzer",
    "time": "2020-02-20T00:00:00.00Z",
    "version": "0"
}
```

Creazione di una regola di evento mediante la console

La procedura seguente descrive come creare una regola di evento utilizzando la console.

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Utilizzando i seguenti valori, crea una EventBridge regola che monitori la ricerca di eventi o acceda agli eventi di anteprima:

   - Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
   - In Event source (Origine eventi), scegli Other (Altro).
   - In Event pattern (Modello di eventi), scegli Custom patterns (JSON editor) (Modelli personalizzati [editor JSON]) e incolla uno dei seguenti esempi di modelli di eventi nell'area di testo:

- Per creare una regola basata su un evento relativo ai risultati degli accessi esterni o inutilizzati, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Access Analyzer Finding"
  ]
}
```

- Per creare una regola basata solo su un evento di accesso ai risultati di accesso non utilizzato, utilizzate il seguente esempio di pattern:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Unused Access Finding for IAM entities"
  ]
}
```

> **Note**
>
> Non è possibile creare una regola basata solo su un evento di accessi esterni.

- Per creare una regola basata su un evento di anteprima di accesso, utilizza il seguente esempio di modello:

```
{
  "source": [
    "aws.access-analyzer"
  ],
  "detail-type": [
    "Access Preview State Change"
  ]
}
```

- Per i tipi di Target, scegli il AWS servizio e per Seleziona una destinazione, scegli una destinazione come un argomento o AWS Lambda una funzione di Amazon SNS. La destinazione viene attivata quando viene ricevuto un evento che corrisponde al modello di evento definito nella regola.

Per ulteriori informazioni sulla creazione di regole, consulta la sezione Creazione di EventBridge regole Amazon che reagiscono agli eventi nella Amazon EventBridge User Guide.

Creazione di una regola evento mediante la CLI

1. Utilizza quanto segue per creare una regola per Amazon EventBridge utilizzando il AWS CLI. Sostituisci il nome della regola *TestRule*con il nome della regola.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-
analyzer\"]}"
```

2. È possibile personalizzare la regola per attivare operazioni di destinazione solo per un sottoinsieme di risultati generati, ad esempio risultati con attributi specifici. Nell'esempio seguente viene illustrato come creare una regola che attiva un'operazione di destinazione solo per i risultati con stato Attivo.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-
analyzer\"],\"detail-type\":[\"Access Analyzer Finding\"],\"detail\":{\"status\":
[\"ACTIVE\"]}}"
```

   Nell'esempio seguente viene illustrato come creare una regola che attiva un'operazione di destinazione solo per le anteprime di accesso con stato da `Creating` a `Completed`.

```
aws events put-rule --name TestRule --event-pattern "{\"source\":[\"aws.access-
analyzer\"],\"detail-type\":[\"Access Preview State Change\"],\"detail\":{\"status
\":[\"COMPLETED\"]}}"
```

3. Per definire una funzione Lambda come destinazione per la regola creata, utilizza il seguente comando di esempio. Sostituire la regione e il nome della funzione nell'ARN come appropriato per l'ambiente in uso.

```
aws events put-targets --rule TestRule --targets Id=1,Arn=arn:aws:lambda:us-
east-1:111122223333:function:MyFunction
```

4. Aggiungere le autorizzazioni necessarie per richiamare la destinazione della regola. Nell'esempio seguente viene illustrato come concedere autorizzazioni a una funzione Lambda seguendo gli esempi precedenti.

```
aws lambda add-permission --function-name MyFunction --statement-id 1 --action
 'lambda:InvokeFunction' --principal events.amazonaws.com
```

Integra Access Analyzer con AWS Security Hub

AWS Security Hubti offre una visione completa dello stato di sicurezza AWS e ti aiuta a controllare il tuo ambiente rispetto agli standard e alle migliori pratiche del settore della sicurezza. Security Hub raccoglie dati sulla sicurezza da tutti AWS gli account, i servizi e i prodotti partner di terze parti supportati e ti aiuta ad analizzare le tendenze in materia di sicurezza e identificare i problemi di sicurezza con la massima priorità.

Quando effettui l'integrazione AWS Identity and Access Management Access Analyzer con Security Hub, puoi inviare i risultati da IAM Access Analyzer a Security Hub. Security Hub può quindi includere tali risultati nella sua analisi della posizione di sicurezza.

Indice

Come Sistema di analisi degli accessi AWS IAM invia i risultati a Security Hub

Nella Centrale di sicurezza, i problemi di sicurezza vengono monitorati come esiti. Alcuni risultati derivano da problemi rilevati da altri AWS servizi o da partner terzi. Security Hub dispone inoltre di una serie di regole che utilizza per rilevare problemi di sicurezza e generare risultati.

Security Hub fornisce strumenti per gestire i risultati da tutte queste fonti. È possibile visualizzare e filtrare gli elenchi di risultati e visualizzare i dettagli per un riscontro. Consulta Visualizzazione dei risultati nella Guida per l'utente di AWS Security Hub . È inoltre possibile monitorare lo stato di un'indagine in un esito. Consulta Operazioni sui risultati nella Guida per l'utente di AWS Security Hub .

Tutti i risultati in Security Hub utilizzano un formato JSON standard chiamato AWS Security Finding Format (ASFF). L'ASFF include dettagli sull'origine del problema, sulle risorse interessate e sullo stato corrente del risultato. Consulta AWS Security Finding Format (ASFF) nella Guida per l'utente di AWS Security Hub .

AWS Identity and Access Management Access Analyzer è uno dei AWS servizi che invia i risultati a Security Hub. Per gli accessi non utilizzati, IAM Access Analyzer rileva l'accesso non utilizzato concesso a utenti o ruoli IAM e genera un risultato per ciascuno di essi. IAM Access Analyzer invia quindi questi risultati a Security Hub. Per quanto riguarda l'accesso esterno, IAM Access Analyzer rileva una dichiarazione di policy che consente l'accesso pubblico o l'accesso tra account a responsabili esterni su una risorsa supportata nell'organizzazione o nell'account. IAM Access Analyzer genera un risultato per l'accesso pubblico, che poi invia a Security Hub. Per l'accesso su più account, IAM Access Analyzer invia un singolo risultato per un principale esterno alla volta a Security Hub. Se sono presenti più risultati tra account in IAM Access Analyzer, è necessario risolvere il risultato del Security Hub per il singolo principale esterno prima che IAM Access Analyzer fornisca la successiva ricerca tra account. Per un elenco completo dei responsabili esterni con accesso su più account al di fuori della zona di fiducia per l'analizzatore, è necessario visualizzare i risultati in IAM Access Analyzer.

Tipi di risultati inviati da Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM invia i risultati a Security Hub utilizzando il formato ASFF (Security Finding Format) di AWS. In ASFF, il `Types` campo fornisce il tipo di esito. I risultati ottenuti da Sistema di analisi degli accessi AWS IAM possono avere i seguenti valori per `Types`.

- Risultati degli accessi esterni: effetti/esposizione dei dati/accesso esterno concesso

- Risultati dell'accesso esterno: controlli del software e della configurazione/Best practice di sicurezza/Accesso esterno garantito AWS
- Risultati relativi agli accessi non utilizzati: controlli del software e della configurazione/best practice di sicurezza/Autorizzazioni non utilizzate AWS
- Risultati relativi agli accessi non utilizzati: controlli del software e della configurazione/best practice di sicurezza/ruolo IAM non utilizzato AWS
- Risultati relativi agli accessi non utilizzati: controlli del software e della configurazione/best practice di sicurezza/password utente IAM non utilizzata AWS
- Risultati di accesso non utilizzati: controlli del software e della configurazione/best practice di sicurezza/chiave di accesso utente IAM non utilizzata AWS

Latenza per l'invio degli esiti

Quando Sistema di analisi degli accessi AWS IAM crea un nuovo risultato, lo invia a Security Hub solitamente entro 30 minuti. In rare occasioni e in determinate condizioni, a Sistema di analisi degli accessi AWS IAM non viene notificato che una policy è stata aggiunta o aggiornata. Ad esempio, una modifica alle impostazioni di accesso pubblico del blocco a livello di account di Amazon S3 può richiedere fino a 12 ore. Inoltre, se si verifica un problema di consegna con la consegna dei AWS CloudTrail log, la modifica della politica non attiva una nuova scansione della risorsa segnalata nel risultato. In questo caso, Sistema di analisi degli accessi AWS IAM analizza la policy nuova o aggiornata durante la scansione periodica successiva.

Nuovo tentativo quando Security Hub non è disponibile

Se Security Hub non è disponibile, Sistema di analisi degli accessi AWS IAM riprova a inviare i risultati su base periodica.

Aggiornamento degli esiti esistenti nella Centrale di sicurezza

Dopo aver inviato un risultato a Security Hub, AWS Identity and Access Management Access Analyzer invia aggiornamenti per riflettere ulteriori osservazioni sull'attività di ricerca a Security Hub. Gli aggiornamenti si riflettono all'interno dello stesso risultato.

Mentre Sistema di analisi degli accessi AWS IAM raggruppa i risultati degli accessi esterni per risorsa, il risultato per una risorsa in Security Hub è attivo se almeno uno dei risultati per la risorsa in Sistema di analisi degli accessi AWS IAM è attivo. Se tutti i risultati in Sistema di analisi degli accessi AWS IAM di una risorsa vengono archiviati o risolti, il risultato di Security Hub viene archiviato. Il risultato di Security Hub viene aggiornato quando si modifica l'accesso alla policy tra l'accesso

pubblico e l'accesso tra account diversi. Questo aggiornamento può includere modifiche al tipo, al titolo, alla descrizione e alla gravità del risultato.

Sistema di analisi degli accessi AWS IAM non raggruppa i risultati degli accessi inutilizzati per risorsa, quindi se un risultato di accesso inutilizzato viene risolto in Sistema di analisi degli accessi AWS IAM, anche il risultato di Security Hub viene risolto. Il risultato di Security Hub viene aggiornato quando aggiorni l'utente o il ruolo IAM che ha generato il risultato di accesso inutilizzato.

Visualizzazione dei risultati di Sistema di analisi degli accessi AWS IAM in Security Hub

Per visualizzare i risultati di Sistema di analisi degli accessi AWS IAM in Security Hub, scegli Visualizza risultati nella sezione AWS: Sistema di analisi degli accessi AWS IAM della pagina di riepilogo. In alternativa, è possibile scegliere Risultati dal pannello di navigazione. È quindi possibile filtrare i risultati per visualizzare solo AWS Identity and Access Management Access Analyzer i risultati scegliendo il campo Nome prodotto: con un valore di**`IAM Access Analyzer`**.

Interpretazione dei nomi dei risultati di Sistema di analisi degli accessi AWS IAM in Security Hub

AWS Identity and Access Management Access Analyzer invia i risultati a Security Hub utilizzando il AWS Security Finding Format (ASFF). In ASFF, il campo Tipi fornisce il tipo di risultato. I tipi ASFF utilizzano uno schema di denominazione diverso da. AWS Identity and Access Management Access Analyzer La tabella seguente include dettagli su tutti i tipi di ASFF associati ai AWS Identity and Access Management Access Analyzer risultati così come appaiono in Security Hub.

Tipo di risultati ASFF	Titolo del risultato di Security Hub	Descrizione
Effetti/Esposizione dei dati/ Accesso esterno concesso	<resource ARN>consente l'accesso pubblico	Una politica basata sulle risorse collegata alla risorsa consente l'accesso pubblico alla risorsa a tutte le entità esterne.
Controlli del software e della configurazione/Best practice di AWS sicurezza/Accesso esterno concesso	<resource ARN> consente l'accesso tra account	Una politica basata sulle risorse collegata alla risorsa consente l'accesso tra account alle entità esterne all'area di trust per l'analizzatore.

Tipo di risultati ASFF	Titolo del risultato di Security Hub	Descrizione
Controlli del software e della configurazione/Best practice di sicurezza/Autorizzazioni non utilizzate AWS	<resource ARN> contiene autorizzazioni inutilizzate	Un utente o un ruolo contiene autorizzazioni di servizio e operazioni inutilizzate.
Controlli del software e della configurazione/Best practice di sicurezza/Ruolo IAM non utilizzato AWS	<resource ARN> contiene un ruolo IAM inutilizzato	Un utente o un ruolo contiene un ruolo IAM inutilizzato.
Controlli AWS del software e della configurazione/ Best practice di sicurezza/ Password utente IAM non utilizzata	<resource ARN> contiene una password utente IAM inutilizz ata	Un utente o un ruolo contiene una password utente IAM inutilizzata.
Controlli del software e della configurazione/Best practice di sicurezza/Chiave di accesso utente IAM AWS non utilizzata	<resource ARN> contiene una chiave di accesso dell'utente IAM inutilizzata	Un utente o un ruolo contiene una chiave di accesso dell'utente IAM inutilizzata.

Risultato tipico da Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM invia i risultati a Security Hub utilizzando AWS Security Finding Format (ASFF).

Ecco un esempio di un tipico risultato da Sistema di analisi degli accessi AWS IAM per i risultati degli accessi esterni.

```
{
    "SchemaVersion": "2018-10-08",
    "Id": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/my-analyzer/
arn:aws:s3:::my-bucket",
    "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/access-analyzer",
    "GeneratorId": "aws/access-analyzer",
    "AwsAccountId": "111122223333",
```

```
    "Types": ["Software and Configuration Checks/AWS Security Best Practices/External
 Access Granted"],
    "CreatedAt": "2020-11-10T16:17:47Z",
    "UpdatedAt": "2020-11-10T16:43:49Z",
    "Severity": {
        "Product": 1,
        "Label": "LOW",
        "Normalized": 1
    },
    "Title": "AwsS3Bucket/arn:aws:s3:::my-bucket/ allows cross-account access",
    "Description": "AWS::S3::Bucket/arn:aws:s3:::my-bucket/ allows cross-account access
 from AWS 444455556666",
    "Remediation": {
        "Recommendation": {"Text": "If the access isn't intended, it indicates a
 potential security risk. Use the console for the resource to modify or remove the
 policy that grants the unintended access. You can use the Rescan button on the Finding
 details page in the Access Analyzer console to confirm whether the change removed the
 access. If the access is removed, the status changes to Resolved."}
    },
    "SourceUrl": "https://console.aws.amazon.com/access-analyzer/home?region=us-
west-2#/findings/details/dad90d5d-63b4-6575-b0fa-ef9c556ge798",
    "Resources": [
        {
            "Type": "AwsS3Bucket",
            "Id": "arn:aws:s3:::my-bucket",
            "Details": {
                "Other": {
                    "External Principal Type": "AWS",
                    "Condition": "none",
                    "Action Granted": "s3:GetObject,s3:GetObjectVersion",
                    "External Principal": "444455556666"
                }
            }
        }
    ],
    "WorkflowState": "NEW",
    "Workflow": {"Status": "NEW"},
    "RecordState": "ACTIVE"
}
```


Ecco un esempio di un tipico risultato da Sistema di analisi degli accessi AWS IAM per i risultati degli accessi inutilizzati.

```
{
    "Findings": [
    {
      "SchemaVersion": "2018-10-08",
      "Id": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/integTestAnalyzer-
DO-NOT-DELETE/arn:aws:iam::111122223333:role/TestRole/UnusedPermissions",
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/access-analyzer",
      "ProductName": "IAM Access Analyzer",
      "CompanyName": "AWS",
      "Region": "us-west-2",
      "GeneratorId": "aws/access-analyzer",
      "AwsAccountId": "111122223333",
      "Types": [
        "Software and Configuration Checks/AWS Security Best Practices/Unused
 Permission"
      ],
      "CreatedAt": "2023-09-18T16:29:09.657Z",
      "UpdatedAt": "2023-09-21T20:39:16.651Z",
      "Severity": {
        "Product": 1,
        "Label": "LOW",
        "Normalized": 1
      },
      "Title": "AwsIamRole/arn:aws:iam::111122223333:role/IsengardRole-DO-NOT-DELETE/
 contains unused permissions",
      "Description": "AWS::IAM::Role/arn:aws:iam::111122223333:role/IsengardRole-DO-
NOT-DELETE/ contains unused service and action-level permissions",
      "Remediation": {
        "Recommendation": {
          "Text":"If the unused permissions aren’t required, delete the permissions to
 refine access to your account. Use the IAM console to modify or remove the policy that
 grants the unused permissions. If all the unused permissions are removed, the status
 of the finding changes to Resolved."
        }
      },
      "SourceUrl": "https://us-west-2.console.aws.amazon.com/access-analyzer/
home?region=us-west-2#/unused-access-findings?resource=arn%3Aaws%3Aiam%3A
%3A903798373645%3Arole%2FTestRole",
      "ProductFields": {
      "numberOfUnusedActions": "256",
      "numberOfUnusedServices": "15",
      "resourceOwnerAccount": "111122223333",
      "findingId": "DEMO24d8d-0d3f-4d3d-99f4-299fc8a62ee7",
```

```
      "findingType": "UnusedPermission",
      "aws/securityhub/FindingId": "arn:aws:securityhub:us-west-2::product/aws/access-
analyzer/arn:aws:access-analyzer:us-west-2:111122223333:analyzer/integTestAnalyzer-DO-
NOT-DELETE/arn:aws:iam::111122223333:role/TestRole/UnusedPermissions",
      "aws/securityhub/ProductName": "AM Access Analyzer",
      "aws/securityhub/CompanyName": "AWS"
    },
    "Resources": [
    {
      "Type": "AwsIamRole",
      "Id": "arn:aws:iam::111122223333:role/TestRole"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
    },
  "RecordState": "ARCHIVED",
  "FindingProviderFields": {
    "Severity": {
      "Label": "LOW"
    },
    "Types": [
    "Software and Configuration Checks/AWS Security Best Practices/Unused Permission"
    ]
  }
  }
]
}
```

Abilitazione e configurazione dell'integrazione

Per utilizzare l'integrazione con Security Hub, è necessario abilitare Security Hub. Per informazioni su come abilitare Security Hub, consulta Configurazione di Security Hub nella Guida per l'utente di AWS Security Hub .

Una volta abilitati Sistema di analisi degli accessi AWS IAM e Security Hub, l'integrazione viene abilitata automaticamente. Sistema di analisi degli accessi AWS IAM inizia immediatamente a inviare i risultati a Security Hub.

Come interrompere l'invio di esiti

Per interrompere l'invio dei risultati a Security Hub, puoi utilizzare la console o l'API di Security Hub.

Consulta Disabilitazione e abilitazione del flusso dei risultati da un'integrazione (console) o Disabilitazione del flusso di risultati da un'integrazione (API Security Hub, AWS CLI) nella Guida per l'utente di AWS Security Hub .

Registrazione delle chiamate API IAM Access Analyzer con AWS CloudTrail

IAM Access Analyzer è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in IAM Access Analyzer. CloudTrail acquisisce tutte le chiamate API per IAM Access Analyzer come eventi. Le chiamate acquisite includono le chiamate dalla console IAM Access Analyzer e le chiamate di codice alle operazioni API di IAM Access Analyzer.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per IAM Access Analyzer. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi.

Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata a IAM Access Analyzer, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su IAM Access Analyzer in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in IAM Access Analyzer, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per IAM Access Analyzer, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati

- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di IAM Access Analyzer vengono registrate CloudTrail e documentate nel riferimento all'API di riferimento di IAM Access Analyzer. Ad esempio, le chiamate a `CreateArchiveRule` e le `CreateAnalyzer` `ListFindings` azioni generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'elemento CloudTrail userIdentity.

Informazioni sulle voci dei file di log di IAM Access Analyzer

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l'`CreateAnalyzer`operazione eseguita da una sessione con ruolo presunto denominata «14 `Alice-tempcreds` giugno 2021". La sessione del ruolo è stata emessa dal ruolo denominato `admin-tempcreds`.

```
{
  "eventVersion": "1.05",
  "userIdentity":   {
    "type": "AssumedRole",
    "principalId": "AROAIBKEVSQ6C2EXAMPLE:Alice-tempcreds",
```

```
    "arn": "arn:aws:sts::111122223333:assumed-role/admin-tempcreds/Alice-tempcreds",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "true",
        "creationDate": "2021-06-14T22:54:20Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin-tempcreds",
        "accountId": "111122223333",
        "userName": "admin-tempcreds"
      },
     "webIdFederationData": {},
    }
  },
  "eventTime": "2021-06-14T22:57:36Z",
  "eventSource": "access-analyzer.amazonaws.com",
  "eventName": "CreateAnalyzer",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.179",
  "userAgent": "aws-sdk-java/1.12.79 Linux/5.4.141-78.230 OpenJDK_64-
Bit_Server_VM/25.302-b08 java/1.8.0_302 vendor/Oracle_Corporation cfg/retry-mode/
standard",
  "requestParameters": {
    "analyzerName": "test",
    "type": "ACCOUNT",
    "clientToken": "11111111-abcd-2222-abcd-222222222222",
        "tags": {
            "tagkey1": "tagvalue1"
        }
  },
  "responseElements": {
    "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/test"
  },
  "requestID": "22222222-dcba-4444-dcba-333333333333",
  "eventID": "33333333-bcde-5555-bcde-444444444444",
  "readOnly": false,
  "eventType": "AwsApiCall",,
  "managementEvent": true,
  "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

Chiavi di filtro di Sistema di analisi degli accessi AWS IAM

Puoi utilizzare le chiavi di filtro seguenti per definire una regola di archiviazione (`CreateArchiveRule`), aggiornare una regola di archiviazione (`UpdateArchiveRule`), recuperare un elenco di risultati (`ListFindings`and `ListFindingsV2`) o recuperare un elenco di risultati di anteprima di accesso per una risorsa (`ListAccessPreviewFindings`). Non c'è differenza tra l'utilizzo dell'API IAM e AWS CloudFormation la configurazione delle regole di archiviazione.

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
risorsa	L'ARN identifica in modo univoco la risorsa a cui l'entità principale esterna ha accesso. Per ulteriori informazioni, consulta Amazon Resource Name (ARN).	Stringa	 Sì	 Sì	 Sì
resourceType `AWS::IAM::Role | AWS::KMS::Key | AWS::Lambda::Function | AWS::Lambda::LayerVersion`	Il tipo di risorsa a cui l'entità principale esterna ha accesso.	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
`| AWS::S3::Bucket | AWS::S3Express::DirectoryBucket | AWS::SQS::Queue | AWS::SecretsManager::Secret | AWS::EFS::FileSystem | AWS::EC2::Snapshot | AWS::ECR::Repository | AWS::RDS::DBSnapshot | AWS::RDS::DBClusterSnapshot | AWS::SNS::Topic |`					

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
`AWS::DynamoDB::Stream` | `AWS::DynamoDB::Table`					
resourceOwnerAccount	L'ID dell' AWS account a 12 cifre che possiede la risorsa. Per ulteriori informazioni, consulta ID account di AWS.	Stringa	 Sì	 Sì	 Sì
isPublic	Indica se il risultato segnala una risorsa che dispone di una policy che consente l'accesso pubblico.	Booleano	 Sì	 Sì	 Sì
Tipo di ricerca `UnusedIAMRole` | `UnusedIAMUserAccessKey` | `UnusedIAMUserPassword` | `UnusedPermission`	Il tipo di risultato . Puoi filtrare solo in base al tipo di ricerca per i risultati di accesso non utilizzati.	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
status `ACTIVE | ARCHIVED | RESOLVED`	Lo stato attuale del risultato.	Stringa	 No	 Sì	 Sì
error	Indica l'errore segnalato per il risultato.	Stringa	 Sì	 Sì	 Sì
principal.AWS	L'account che ha concesso l'accesso alla risorsa nel campo `Principal` dell'esito. Inserisci l'ID AWS account a 12 cifre o l'ARN dell'utente o del ruolo esterno AWS . Per ulteriori informazioni, consulta ID account di AWS.	Stringa	 Sì	 Sì	 Sì
principal.Federated	L'ARN dell'identità federata che ha accesso alla risorsa nel risultato . Per ulteriori informazioni, consulta Provider di identità e federazione	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
condizione.aws: PrincipalArn	L'ARN del principale (utente IAM, ruolo o gruppo) indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	 Sì	 Sì	 Sì
condition.aws: IDPrincipalOrg	L'identificatore dell'organizzazione dell'entità principale indicata come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	 Sì	 Sì	 Sì
condizione.aws: PrincipalOrgPaths	L'ID dell'organizzazione o dell'unità organizzativa (OU) indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
condizione.aws:SourceIp	L'indirizzo IP che consente all'entità principale di accedere alla risorsa quando si utilizza l'indirizzo IP specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Indirizzo IP	 Sì	 Sì	 Sì
condizione.aws:SourceVpc	L'ID VPC che consente l'accesso dell'entità principale alla risorsa quando si utilizza il VPC specificato. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	 Sì	 Sì	 Sì
condizione.aws:UserId	L'ID utente dell'utente IAM da un account esterno indicato come condizione per l'accesso alla risorsa. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
condition .cognito-identity. amazonaws .com:aud	L'ID pool di identità di Amazon Cognito specificato come condizione per l'accesso al ruolo IAM nel risultato. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	 Sì	 Sì	 Sì
condition .graph.facebook.com:app_id	L'ID dell'applicazione Facebook (o l'ID del sito) specificato come condizione per consentire l'accesso con la federazione Accedi con Facebook al ruolo IAM nel risultato. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	 Sì	 Sì	 Sì
condition .accounts .google.com:aud	L'ID dell'applicazione Google specificato come condizione per l'accesso al ruolo IAM. Per saperne di più, consulta IAM e AWS STS condition context keys.	Stringa	 Sì	 Sì	 Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
condizione.kms:CallerAccount	L'ID AWS dell'account proprietario dell'entità chiamante (utente IAM, ruolo o utente root dell'account) utilizzata dalle chiamate ai servizi. AWS KMS Per ulteriori informazioni, consulta Condition keys for AWS Key Management Service.	Stringa	Sì	Sì	Sì
condition.www.amazon.com:app_id	L'ID dell'applicazione Amazon (o l'ID del sito) specificato come condizione per consentire l'accesso con la federazione Login with Amazon al ruolo. Per ulteriori informazioni, consulta la sezione	Stringa	Sì	Sì	Sì
id	L'ID del risultato.	Stringa	No	Sì	Sì

Criterion	Descrizione	Type	Regola di archivio	Elenco di risultati	Visualizzazione dei risultati di anteprima accesso
changeType	Fornisce un contesto sul modo in cui il risultato dell'anteprima di accesso viene confrontato con l'accesso identificato esistente in Sistema di analisi degli accessi AWS IAM.	Stringa	 No	 No	 Sì
existingFindingId	L'ID esistente del risultato in Sistema di analisi degli accessi AWS IAM, fornito solo per i risultati esistenti nell'anteprima dell'accesso.	Stringa	 No	 No	 Sì
existingFindingStatus	Lo stato esistente del risultato in Access Analyzer, fornito solo per i risultati esistenti nell'anteprima dell'accesso.	Stringa	 No	 No	 Sì

Utilizzo di ruoli collegati ai servizi per AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer utilizza un ruolo collegato al servizio IAM. Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente ad Sistema di analisi degli accessi AWS IAM. I ruoli collegati ai servizi sono predefiniti da IAM Access Analyzer e includono tutte le autorizzazioni richieste dalla funzionalità per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Sistema di analisi degli accessi AWS IAM perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Sistema di analisi degli accessi AWS IAM definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, solo Sistema di analisi degli accessi AWS IAM potrà assumere i relativi ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per AWS Identity and Access Management Access Analyzer

AWS Identity and Access Management Access Analyzer utilizza il ruolo collegato ai servizi denominato AWSServiceRoleForAccessAnalyzer— Allow Access Analyzer per analizzare i metadati delle risorse per l'accesso esterno e analizzare le attività per identificare gli accessi non utilizzati.

Il ruolo AWSServiceRoleForAccessAnalyzer collegato ai servizi prevede che i seguenti servizi assumano il ruolo:

- `access-analyzer.amazonaws.com`

La policy delle autorizzazioni del ruolo denominata `AccessAnalyzerServiceRolePolicy` consente ad Sistema di analisi degli accessi AWS IAM di completare le operazioni riportate di seguito su risorse specifiche.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per Sistema di analisi degli accessi AWS IAM

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando abiliti Access Analyzer nell'API AWS Management Console o nell' AWS API, IAM Access Analyzer crea automaticamente il ruolo collegato al servizio. Lo stesso ruolo collegato ai servizi viene utilizzato in tutte le regioni in cui abiliti Sistema di analisi degli accessi AWS IAM. Sia l'accesso esterno che i risultati degli accessi inutilizzati utilizzano lo stesso ruolo collegato al servizio.

> **Note**
>
> Sistema di analisi degli accessi AWS IAM è un servizio regionale. Devi abilitare Sistema di analisi degli accessi AWS IAM in ogni regione in modo indipendente.

Se elimini questo ruolo collegato ai servizi, Sistema di analisi degli accessi AWS IAM crea nuovamente il ruolo alla successiva creazione di un analizzatore.

Puoi utilizzare la console IAM anche per creare un ruolo collegato ai servizi con il caso d'uso Access Analyzer. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `access-analyzer.amazonaws.com` Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, puoi utilizzare lo stesso processo per crearlo nuovamente.

Modifica di un ruolo collegato ai servizi per Sistema di analisi degli accessi AWS IAM

IAM Access Analyzer non consente di modificare il AWSServiceRoleForAccessAnalyzer ruolo collegato al servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Sistema di analisi degli accessi AWS IAM

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non hai un'entità non utilizzata che non viene monitorata o gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

> **Note**
>
> Se Sistema di analisi degli accessi AWS IAM sta utilizzando il ruolo quando provi a eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse IAM Access Analyzer utilizzate da AWSServiceRoleForAccessAnalyzer

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nella sezione Access reports (Report di accesso) in Access analyzer (Analizzatore accesso) scegliere Analyzers (Analizzatori).
3. Selezionare la casella di controllo in alto a sinistra sopra l'elenco degli analizzatori nella tabella Analyzers (Analizzatori) per selezionare tutti gli analizzatori.
4. Scegli Elimina.
5. Per confermare l'eliminazione dell'analizzatore, immettere **`delete`**, quindi scegliere Delete (Elimina).

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al AWSServiceRoleForAccessAnalyzer servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Regioni ed endpoint di AWS.

Anteprima dell'accesso

Oltre ad aiutarti a identificare le risorse condivise con un'entità esterna, AWS IAM Access Analyzer ti mostra anche un'anteprima dei risultati di IAM Access Analyzer prima di distribuire le autorizzazioni per le risorse, in modo da poter verificare che le modifiche alle policy garantiscano solo l'accesso pubblico e interaccount previsto alla tua risorsa. In questo modo è possibile iniziare con l'accesso esterno previsto alle risorse.

È possibile visualizzare in anteprima e convalidare l'accesso pubblico e tra account ai propri bucket Amazon S3 nella sezione Console Amazon S3. Puoi anche utilizzare le API IAM Access Analyzer per visualizzare in anteprima l'accesso pubblico e tra account per i tuoi bucket Amazon S3, AWS KMS le chiavi, i ruoli IAM, le code Amazon SQS e i segreti di Secrets Manager fornendo le autorizzazioni proposte per la tua risorsa.

Argomenti

- Visualizzazione in anteprima dell'accesso nella console Amazon S3
- Anteprima dell'accesso con le API di IAM Access Analyzer

Visualizzazione in anteprima dell'accesso nella console Amazon S3

Dopo aver completato la policy del bucket nella console Amazon S3, è possibile visualizzare in anteprima l'accesso multi-account e pubblico al proprio bucket Amazon S3. È possibile verificare che le modifiche alle policy concedano solo l'accesso esterno previsto prima di scegliere Salva modifiche. Questo passaggio facoltativo ti consente di visualizzare in anteprima AWS Identity and Access Management Access Analyzer i risultati per il tuo bucket. È possibile verificare se la modifica alle policy introduce nuovi risultati o risolve i risultati esistenti per l'accesso esterno. È possibile saltare questa fase di convalida e salvare la propria policy del bucket Amazon S3 in qualsiasi momento.

Per visualizzare in anteprima l'accesso esterno al bucket, è necessario disporre di un analizzatore account attivo nell'area del bucket con l'account come zona di attendibilità. Devi inoltre disporre delle autorizzazioni necessarie per utilizzare IAM Access Analyzer e l'accesso in anteprima. Per ulteriori informazioni sull'abilitazione di IAM Access Analyzer e sulle autorizzazioni richieste, consulta Abilitazione di Sistema di analisi degli accessi AWS IAM.

Come visualizzare in anteprima l'accesso al bucket Amazon S3 quando si crea o si modifica la policy del bucket

1. Una volta completata la creazione o la modifica della policy del bucket, accertarsi che la policy sia una policy del bucket Amazon S3 valida. L'ARN della policy deve corrispondere all'ARN del bucket e gli elementi delle policy devono essere validi.
2. Sotto la policy, in Anteprima accesso esterno, selezionare un analizzatore di account attivo, quindi scegliere Anteprima. Viene generata un'anteprima dei risultati di IAM Access Analyzer per il bucket. L'anteprima analizza la policy del bucket Amazon S3 visualizzata insieme alle autorizzazioni del bucket esistenti. Ciò include le impostazioni relative al bucket e al BPA dell'account, l'ACL del bucket, i punti di accesso Amazon S3 e i punti di accesso multi-regione collegati al bucket con le relative policy e impostazioni BPA.
3. Al termine dell'anteprima di accesso, viene visualizzata un'anteprima dei risultati di IAM Access Analyzer. Ogni risultato riporta un'istanza di un principale esterno all'account con accesso al bucket dopo aver salvato la policy. È possibile convalidare l'accesso al bucket esaminando ogni risultato. L'intestazione del risultato fornisce un riepilogo dell'accesso ed è possibile espanderla per esaminare i dettagli del risultato. I badge del risultato forniscono un contesto su come il salvataggio della policy del bucket cambierebbe l'accesso al bucket. Ad esempio, consentono

verificare se la modifica alle policy introduce nuovi risultati o risolve i risultati esistenti per l'accesso esterno.

a. Novità: indica un risultato per un nuovo accesso esterno che la policy introdurrebbe.
b. Risolto: indica un risultato per l'accesso esterno esistente che la policy rimuoverebbe.
c. Archiviato: indica un risultato per un nuovo accesso esterno che verrebbe archiviato automaticamente in base alle regole di archiviazione per l'analizzatore che definiscono quando i risultati devono essere contrassegnati come previsto.
d. Esistente: indica un risultato esistente per l'accesso esterno che rimarrebbe invariato.
e. Pubblico: se un risultato è per l'accesso pubblico alla risorsa, oltre a uno dei badge precedenti avrà un badge Pubblico.

4. Se si identifica un accesso esterno che non si desidera introdurre o rimuovere, è possibile modificare la policy e scegliere di nuovo Anteprima fino a quando non si raggiunge l'accesso esterno desiderato. Se si dispone di un risultato etichettato come Pubblico, si consiglia di rivedere la policy per rimuovere l'accesso pubblico prima di scegliere Salva modifiche. L'anteprima dell'accesso è una fase facoltativa ed è possibile scegliere Salva modifiche in qualsiasi momento.

Anteprima dell'accesso con le API di IAM Access Analyzer

Puoi utilizzare le API IAM Access Analyzer per visualizzare in anteprima l'accesso pubblico e tra account per i tuoi bucket Amazon S3, AWS KMS le chiavi, i ruoli IAM, le code Amazon SQS e i segreti di Secrets Manager. È possibile visualizzare in anteprima l'accesso fornendo le autorizzazioni proposte per una risorsa esistente di cui si è proprietari o per una nuova risorsa che si desidera implementare.

Per visualizzare in anteprima l'accesso esterno alla risorsa, è necessario disporre di un analizzatore account attivo per l'account e la regione della risorsa. Devi inoltre disporre delle autorizzazioni necessarie per utilizzare IAM Access Analyzer e l'accesso in anteprima. Per ulteriori informazioni sull'abilitazione di IAM Access Analyzer e sulle autorizzazioni richieste, consulta Abilitazione di Sistema di analisi degli accessi AWS IAM.

Per visualizzare in anteprima l'accesso a una risorsa, è possibile utilizzare l'operazione `CreateAccessPreview` e fornire l'ARN dell'analizzatore e la configurazione del controllo degli accessi per la risorsa. Il servizio restituisce l'ID univoco per l'anteprima di accesso che è possibile utilizzare per verificare lo stato dell'anteprima di accesso con l'operazione `GetAccessPreview`. Quando lo stato è `Completed`, è possibile utilizzare l'operazione `ListAccessPreviewFindings`

per recuperare i risultati generati per l'anteprima dell'accesso. Le operazioni `GetAccessPreview` e `ListAccessPreviewFindings` recupereranno le anteprime di accesso e i risultati creati entro circa 24 ore.

Ogni risultato recuperato contiene i dettagli del risultato che descrivono l'accesso. Uno stato di anteprima del risultato che descrive se il risultato sarebbe `Active`, `Archived` oppure `Resolved` dopo l'implementazione delle autorizzazioni e un `changeType`. Il `changeType` fornisce un contesto sul modo in cui il risultato dell'anteprima di accesso viene confrontato con l'accesso esistente identificato in IAM Access Analyzer.

- Novità: il risultato riguarda l'accesso appena introdotto.
- Invariato: il risultato dell'anteprima è un risultato esistente che rimarrebbe invariato.
- Modificato: il risultato dell'anteprima è un risultato esistente con un cambiamento di stato.

Le operazioni `status` e `changeType` consentono di capire come la configurazione delle risorse modificherebbe l'accesso delle risorse esistenti. Se `changeType` è `Unchanged` o Modificato, il risultato conterrà anche l'ID e lo stato del risultato esistenti in IAM Access Analyzer. Ad esempio, un risultato `Changed` con stato di anteprima `Resolved` e stato esistente `Active` indica che il risultato `Active` esistente per la risorsa diventerebbe `Resolved` in seguito alla modifica delle autorizzazioni proposta.

È possibile utilizzare l'operazione `ListAccessPreviews` per recuperare un elenco di anteprime di accesso per l'analizzatore specificato. Questa operazione recupererà le informazioni sull'anteprima di accesso creata in un'ora circa.

In generale, se l'anteprima di accesso è per una risorsa esistente e si lascia un'opzione di configurazione non specificata, l'anteprima di accesso utilizzerà la configurazione della risorsa esistente per impostazione predefinita. Se l'anteprima di accesso è relativa a una nuova risorsa e si lascia un'opzione di configurazione non specificata, l'anteprima di accesso utilizzerà il valore predefinito in base al tipo di risorsa. Per i casi di configurazione per ciascun tipo di risorsa, consultare gli argomenti di seguito.

Visualizzazione in anteprima dell'accesso al bucket Amazon S3

Per creare un'anteprima di accesso per un nuovo bucket Amazon S3 o per un proprio bucket Amazon S3 esistente, è possibile proporre una configurazione del bucket specificando la policy del bucket di Amazon S3, gli ACL del bucket, le impostazioni del BPA del bucket e i punti di accesso Amazon S3, inclusi i punti di accesso multi-regione, collegati al bucket.

> **Note**
>
> Prima di provare a creare un'anteprima di accesso per un nuovo bucket, ti consigliamo di chiamare l'operazione Amazon HeadBucketS3 per verificare se il bucket denominato esiste già. Questa operazione è utile per determinare se esiste un bucket e si dispone dell'autorizzazione per accedervi.

Policy del bucket: se la configurazione è per un bucket Amazon S3 esistente e non si specifica la policy del bucket Amazon S3, l'anteprima di accesso utilizza la policy esistente allegata al bucket. Se l'anteprima di accesso riguarda una nuova risorsa e non si specifica la policy del bucket di Amazon S3, l'anteprima di accesso presuppone un bucket senza policy. Per proporre l'eliminazione di una policy del bucket esistente, è possibile specificare una stringa vuota. Per ulteriori informazioni sui limiti delle policy del bucket supportati, consultare Esempi di policy di bucket.

Concessioni ACL del bucket: è possibile proporre fino a 100 concessioni ACL per bucket. Se la configurazione di concessione proposta è per un bucket esistente, l'anteprima dell'accesso utilizza l'elenco proposto di configurazioni di concessioni al posto delle concessioni esistenti. In caso contrario, utilizzerà le concessioni esistenti per il bucket.

Punti di accesso del bucket: l'analisi supporta fino a 100 punti di accesso, inclusi i punti di accesso multi-regione, per bucket, con un massimo di dieci nuovi punti di accesso che è possibile proporre per bucket. Se la configurazione dei punti di accesso di Amazon S3 proposta è per un bucket esistente, l'anteprima dell'accesso utilizza la configurazione dei punti di accessi proposta al posto dei punti di accesso esistenti. Per proporre un punto di accesso senza una policy, è possibile fornire una stringa vuota come policy del punto di accesso. Per ulteriori informazioni sui limiti delle policy dei punti di accesso, consultare Restrizioni e limitazioni dei punti di accesso.

Configurazione dell'accesso pubblico ai blocchi: se la configurazione proposta è per un bucket Amazon S3 esistente e non si specifica la configurazione, l'anteprima dell'accesso utilizza l'impostazione esistente. Se la configurazione proposta riguarda un nuovo bucket e non si specifica la configurazione BPA del bucket, l'anteprima dell'accesso utilizza `false`. Se la configurazione proposta si riferisce a un nuovo punto di accesso o a un punto di accesso multi-regione e non si specifica la configurazione BPA del punto di accesso, l'anteprima dell'accesso utilizza `true`.

Visualizzazione in anteprima dell'accesso alla chiave AWS KMS

Per creare un'anteprima di accesso per una nuova AWS KMS chiave o per una AWS KMS chiave esistente di tua proprietà, puoi proporre una configurazione delle chiavi specificando la politica delle AWS KMS chiavi e la configurazione della concessione. AWS KMS

AWS KMS politica chiave: se la configurazione riguarda una chiave esistente e non si specifica la politica della chiave, l'anteprima di accesso utilizza la politica esistente per la chiave. Se l'anteprima di accesso è relativa a una nuova risorsa e non si specifica la policy della chiave, l'anteprima di accesso utilizza la policy della chiave predefinita. La chiave della policy proposta non può essere una stringa vuota.

AWS KMS sovvenzioni: l'analisi supporta fino a 100 concessioni KMS per configurazione*.* Se la configurazione di concessione proposta riguarda una chiave esistente, l'anteprima di accesso utilizza l'elenco proposto di configurazioni di concessione al posto delle sovvenzioni esistenti. In caso contrario, utilizzerà le concessioni esistenti per la chiave.

Anteprima dell'accesso al ruolo IAM

Per creare un'anteprima di accesso per un nuovo ruolo IAM o un ruolo IAM esistente, puoi proporre una configurazione del ruolo IAM specificando la policy di attendibilità.

Policy di attendibilità del ruolo: se la configurazione riguarda un nuovo ruolo IAM, è necessario specificare la policy di attendibilità. Se la configurazione riguarda un ruolo IAM esistente di cui si è proprietari e non propone la policy di attendibilità, l'anteprima di accesso utilizza la policy di attendibilità esistente per il ruolo. La policy proposta non può essere una stringa vuota.

Anteprima dell'accesso alla coda Amazon SQS

Per creare un'anteprima di accesso per una nuova coda Amazon SQS o una coda Amazon SQS esistente di proprietà, è possibile proporre una configurazione di coda Amazon SQS specificando la policy di Amazon SQS per la coda.

Policy della coda Amazon SQS: se la configurazione è per una coda Amazon SQS esistente e non si specifica la policy di Amazon SQS, l'anteprima degli accessi utilizza la policy di Amazon SQS esistente per la coda. Se l'anteprima di accesso riguarda una nuova risorsa e non specifichi la policy, l'anteprima di accesso presuppone una coda Amazon SQS senza policy. Per proporre l'eliminazione di una policy di coda Amazon SQS esistente, è possibile specificare una stringa vuota per la policy di Amazon SQS.

Anteprima dell'accesso al segreto di Secrets Manager

Per creare un'anteprima di accesso per un nuovo segreto di Secrets Manager o un segreto di Secrets Manager esistente di tua proprietà, puoi proporre una configurazione segreta di Secrets Manager specificando la policy segreta e la chiave di AWS KMS crittografia opzionale.

Policy del segreto: se la configurazione è per un segreto esistente e non specifichi la policy del segreto, l'anteprima di accesso utilizza la policy esistente per il segreto. Se l'anteprima di accesso riguarda una nuova risorsa e non specifichi la policy, l'anteprima di accesso presuppone un segreto senza policy. Per proporre l'eliminazione di una policy esistente, puoi specificare una stringa vuota.

AWS KMS chiave di crittografia: se la configurazione proposta riguarda un nuovo segreto e non si specifica l'ID della AWS KMS chiave, l'anteprima di accesso utilizza la chiave KMS predefinita dell' AWS account. Se si specifica una stringa vuota per l'ID della AWS KMS chiave, l'anteprima di accesso utilizza la chiave KMS predefinita dell' AWS account.

Controlli per la convalida delle policy

Sistema di analisi degli accessi AWS IAM fornisce controlli delle policy che aiutano a convalidare le policy IAM prima di collegarle a un'entità. Questi includono i controlli base forniti dalla convalida delle policy per convalidare la policy rispetto alla sintassi e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy.

Puoi utilizzare i controlli delle policy personalizzati per verificare la presenza di nuovi accessi in base ai tuoi standard di sicurezza. Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Argomenti

- Convalida delle policy di Sistema di analisi degli accessi AWS IAM
- Controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM

Convalida delle policy di Sistema di analisi degli accessi AWS IAM

Puoi convalidare le tue politiche utilizzando la convalida AWS Identity and Access Management Access Analyzer delle politiche. Puoi creare o modificare una policy utilizzando l' AWS API o AWS CLI l'editor di policy JSON nella console IAM. Sistema di analisi degli accessi AWS IAM convalida la

policy rispetto alla sintassi delle policy IAM e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono suggerimenti utili che consentono di creare policy funzionali e conformi alle best practice per la sicurezza. Per visualizzare un elenco dei controlli delle policy di base eseguiti da Sistema di analisi degli accessi AWS IAM, consulta Riferimento ai controlli delle policy di Access Analyzer.

Convalida delle policy in IAM (console)

È possibile visualizzare i risultati generati dalla convalida delle policy di Sistema di analisi degli accessi AWS IAM quando si crea o si modifica una policy gestita nella console IAM. È inoltre possibile visualizzare questi risultati per le policy di utenti o ruoli in linea. Sistema di analisi degli accessi AWS IAM non genera questi risultati per le policy di gruppo in linea.

Come visualizzare i risultati generati dai controlli delle policy per le policy JSON IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Inizia a creare o modificare una policy utilizzando uno dei seguenti metodi:
   a. Per creare una nuova policy gestita, visita la pagina Policy e crea una nuova policy. Per ulteriori informazioni, consulta Creazione di policy utilizzando l'editor JSON.
   b. Per visualizzare i controlli della policy per una policy gestita dal cliente esistente, accedi alla pagina Policy, scegli il nome di una policy, quindi scegli Modifica. Per ulteriori informazioni, consulta Modifica di policy gestite dal cliente (console).
   c. Per visualizzare i controlli della policy relativi a una policy inline per un utente o un ruolo, accedi alla pagina Utenti o Ruoli, scegli il nome di un utente o di un ruolo, scegli il nome della policy nella scheda Autorizzazioni, quindi scegli Modifica. Per ulteriori informazioni, consulta Modifica di policy gestite dal cliente (console).
3. Sceglie la scheda JSON nell'editor di policy.
4. Nel riquadro di convalida delle policy sotto la policy scegli una o più schede delle seguenti opzioni. I nomi delle schede indicano anche il numero di ciascun tipo di ricerca per la policy.
   - Sicurezza: visualizza gli avvisi se la tua policy consente l'accesso che AWS considera un rischio per la sicurezza perché l'accesso è eccessivamente permissivo.
   - Errori: consente di visualizzare gli errori se la policy include righe che impediscono il funzionamento della policy.

- Avvisi: visualizza gli avvisi se le policy non sono conformi alle best practice, ma i problemi non costituiscono rischi per la sicurezza.
- Suggerimenti: visualizza i suggerimenti se AWS consiglia miglioramenti che non influenzano le autorizzazioni delle policy.

5. Esamina i dettagli della ricerca forniti dal controllo delle policy di Sistema di analisi degli accessi AWS IAM. Ogni risultato indica la posizione del problema segnalato. Per ulteriori informazioni sulle cause del problema e su come risolverlo, scegli il collegamento Ulteriori informazioni accanto al risultato. È inoltre possibile ricercare il controllo delle policy associato a ogni ricerca nella pagina di riferimento Controlli delle policy di Access Analyzer.
6. Facoltativo. Se stai modificando una policy esistente, puoi eseguire un controllo personalizzato delle policy per determinare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente. Nel riquadro di convalida delle policy sotto la policy, scegli la scheda Verifica nuovi accessi e seleziona Verifica policy. Se le autorizzazioni modificate concedono un nuovo accesso, l'istruzione verrà evidenziata nel riquadro di convalida della policy. Se non intendi concedere un nuovo accesso, aggiorna le istruzioni di policy e scegli Verifica policy finché non viene rilevato alcun nuovo accesso. Per ulteriori informazioni, consulta Controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM.

   > **Note**
   >
   > Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

7. Aggiorna la policy per risolvere i risultati.

   > **Important**
   >
   > Esegui accuratamente il test delle policy nuove o modificate prima di implementarle nel flusso di lavoro di produzione.

8. Quando hai terminato, seleziona Successivo. Il Validatore di policy segnala eventuali errori di sintassi non riportati da Sistema di analisi degli accessi AWS IAM.

   > **Note**
   >
   > È possibile alternare le schede Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nella scheda Visivo, IAM potrebbe

ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

9. Per le nuove policy, nella pagina Rivedi e crea, immettere un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi le Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona Create policy (Crea policy) per salvare il proprio lavoro.

   Per le policy esistenti, nella pagina Rivedi e crea, controlla le Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona la casella Imposta questa nuova versione come predefinita per salvare la versione aggiornata come versione predefinita della policy. Quindi scegli Salva le modifiche per salvare il lavoro.

Convalida delle policy utilizzando IAM Access Analyzer (o API)AWS CLIAWS

È possibile visualizzare i risultati generati dalla convalida delle policy di Sistema di analisi degli accessi AWS IAM da AWS Command Line Interface (AWS CLI).

Per visualizzare i risultati generati dalla convalida delle policy (o API) di IAM Access Analyzer AWS CLIAWS

Utilizzare una delle seguenti operazioni:

- AWS CLI: aws accessanalyzer validate-policy
- AWS API: ValidatePolicy

Riferimento ai controlli delle policy di Access Analyzer

Puoi convalidare le tue politiche utilizzando la convalida AWS Identity and Access Management Access Analyzer delle politiche. Puoi creare o modificare una policy utilizzando l' AWS API o AWS CLI l'editor di policy JSON nella console IAM. Sistema di analisi degli accessi AWS IAM convalida la policy rispetto alla sintassi delle policy IAM e alle best practice AWS. È possibile visualizzare i risultati del controllo della convalida delle policy che includono avvisi di sicurezza, errori, avvisi generali e suggerimenti per la policy. Questi risultati forniscono suggerimenti utili che consentono di creare policy funzionali e conformi alle best practice per la sicurezza. L'elenco dei controlli di base delle policy forniti da Sistema di analisi degli accessi AWS IAM è disponibile di seguito. L'esecuzione dei controlli di convalida delle policy non comporta costi aggiuntivi. Per ulteriori informazioni sulla

convalida delle policy tramite l'apposito procedimento, consulta Convalida delle policy di Sistema di analisi degli accessi AWS IAM.

Errore: account ARN non consentito

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
ARN account not allowed: The service {{service}} does not support specifying an account
 ID in the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not support specifying an account ID in
 the resource ARN."
```

Risoluzione dell'errore

Rimuovi l'ID account dall'ARN della risorsa. Gli ARN delle risorse per alcuni AWS servizi non supportano la specificazione di un ID di account.

Ad esempio, Amazon S3 non supporta un ID account come spazio dei nomi negli ARN del bucket. Il nome di un bucket Amazon S3 è unico a livello globale e lo spazio dei nomi è condiviso da tutti gli account. AWS Per visualizzare tutti i tipi di risorse disponibili in Amazon S3, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Termini correlati

- Risorse relative alle policy
- Identificatori account
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN non consentita

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
ARN Region not allowed: The service {{service}} does not support specifying a Region in
 the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not support specifying a Region in the
 resource ARN."
```

Risoluzione dell'errore

Rimuovi la regione dall'ARN della risorsa. Gli ARN delle risorse per alcuni AWS servizi non supportano la specificazione di una regione.

Ad esempio, IAM è un servizio globale. La parte della regione di un ARN della risorsa IAM viene sempre mantenuta vuota. Le risorse IAM sono globali, come lo è oggi un AWS account. Ad esempio, dopo aver effettuato l'accesso come utente IAM, puoi accedere ai AWS servizi in qualsiasi area geografica.

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: mancata corrispondenza del tipo di dati

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Data type mismatch: The text does not match the expected JSON data type {{data_type}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The text does not match the expected JSON data type {{data_type}}."
```

Risoluzione dell'errore

Aggiorna il testo per utilizzare il tipo di dati supportato.

Ad esempio, la chiave di condizione globale di `Version` richiede un tipo di dati `String`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione

Errore: chiavi duplicate con un diverso formato maiuscolo/minuscolo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Duplicate keys with different case: The condition key {{key}} appears more than once
 with different capitalization in the same condition block. Remove the duplicate
 condition keys.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} appears more than once with different
 capitalization in the same condition block. Remove the duplicate condition keys."
```

Risoluzione dell'errore

Esamina le chiavi di condizione simili all'interno dello stesso blocco di condizione e utilizza lo stesso formato maiuscolo/minuscolo per tutte le istanze.

Un blocco di condizione è il testo all'interno dell'elemento `Condition` di una istruzione della policy. I nomi delle chiavi di condizione non distinguono tra maiuscole e minuscole. La distinzione tra maiuscole e minuscole dei valori delle chiavi di condizione dipende dall'operatore di condizione utilizzato. Per ulteriori informazioni sul formato maiuscolo/minuscolo per le chiavi di condizione, consulta Elementi delle policy JSON IAM: Condition.

Termini correlati

- Condizioni
- Blocco di condizione
- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: operazione non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid action: The action {{action}} does not exist. Did you mean {{valid_action}}?
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} does not exist. Did you mean
 {{valid_action}}?"
```

Risoluzione dell'errore

L'azione specificata non è valida. Ciò può verificarsi se si digita male il prefisso del servizio o il nome dell'operazione. Per alcuni problemi comuni, il controllo delle policy restituisce un'operazione suggerita.

Termini correlati

- Azioni di policy
- AWS azioni di servizio

AWS politiche gestite con questo errore

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono azioni non valide nelle relative dichiarazioni di policy. Le operazioni non valide non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una politica AWS gestita come riferimento per creare una politica gestita, si AWS consiglia di rimuovere le azioni non valide dalla politica.

- AmazonEMR _v2 FullAccessPolicy
- CloudWatchSyntheticsFullAccess

Errore. account ARN non valido

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Invalid ARN account: The resource ARN account ID {{account}} is not valid. Provide a
 12-digit account ID.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN account ID {{account}} is not valid. Provide a 12-digit account ID."
```

Risoluzione dell'errore

Aggiorna l'ID account nell'ARN della risorsa. Gli ID account sono numeri interi a 12 cifre. Per informazioni su come visualizzare l'ID del tuo account, vedi Trovare l'ID AWS del tuo account.

Termini correlati

- Risorse relative alle policy
- Identificatori account
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: prefisso ARN non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN prefix: Add the required prefix (arn) to the resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add the required prefix (arn) to the resource ARN."
```

Risoluzione dell'errore

AWS gli ARN delle risorse devono includere il prefisso richiesto. `arn:`

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN Region: The Region {{region}} is not valid for this resource. Update the
 resource ARN to include a supported Region.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Region {{region}} is not valid for this resource. Update the
 resource ARN to include a supported Region."
```

Risoluzione dell'errore

Il tipo di risorsa non è supportato nella regione specificata. Per una tabella dei AWS servizi supportati in ogni regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- Nomi e codici delle regioni

Errore: risorsa ARN non valida

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Invalid ARN resource: Resource ARN does not match the expected ARN format. Update the
 resource portion of the ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Resource ARN does not match the expected ARN format. Update the
 resource portion of the ARN."
```

Risoluzione dell'errore

L'ARN della risorsa deve corrispondere alle specifiche per i tipi di risorse noti. Per visualizzare il formato ARN previsto per un servizio, vedere Azioni, risorse e chiavi di condizione per i AWS servizi. Sceglie il nome del servizio per visualizzarne i tipi di risorse e i formati ARN.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: caso di servizio ARN non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid ARN service case: Update the service name ${service} in the resource ARN to use
 all lowercase letters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Update the service name ${service} in the resource ARN to use all
 lowercase letters."
```

Risoluzione dell'errore

Il servizio nell'ARN della risorsa deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: tipo di dati di condizione non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition data type: The condition value data types do not match. Use condition
 values of the same JSON data type.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition value data types do not match. Use condition values of
 the same JSON data type."
```

Risoluzione dell'errore

Il valore nella coppia chiave-valore condizione deve corrispondere al tipo di dati della chiave di condizione e dell'operatore di condizione. Per visualizzare il tipo di dati della chiave di condizione per un servizio, vedi Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

Ad esempio, la chiave di condizione globale di `CurrentTime` supporta l'operatore di condizione `Date`. Se si specifica una stringa o un numero intero per il valore nel blocco di condizione, il tipo di dati non corrisponderà.

Termini correlati

- Condizioni
- Blocco di condizione
- Elementi della policy JSON IAM: operatori di condizione
- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: formato della chiave di condizione non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition key format: The condition key format is not valid. Use the format
 service:keyname.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key format is not valid. Use the format
 service:keyname."
```

Risoluzione dell'errore

La chiave nella coppia chiave-valore condizione deve corrispondere alle specifiche del servizio. Per visualizzare le chiavi di condizione per un servizio, vedi Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

Termini correlati

- Condizioni
- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: booleano multiplo della condizione non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition multiple Boolean: The condition key does not support multiple Boolean
 values. Use a single Boolean value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key does not support multiple Boolean values. Use a
 single Boolean value."
```

Risoluzione dell'errore

La chiave nella coppia chiave-valore della condizione prevede un singolo valore booleano. Quando si forniscono più valori booleani, la corrispondenza della condizione potrebbe non restituire i risultati previsti.

Per visualizzare le chiavi di condizione per un servizio, vedi Azioni, risorse e chiavi di condizione per AWS i servizi. Scegli il nome del servizio per visualizzarne le chiavi di condizione.

- Condizioni

- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: operatore di condizione non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid condition operator: The condition operator {{operator}} is not valid. Use a
 valid condition operator.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition operator {{operator}} is not valid. Use a valid
 condition operator."
```

Risoluzione dell'errore

Aggiorna la condizione per utilizzare un operatore di condizione supportato.

Termini correlati

- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Errore: effetto non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid effect: The effect {{effect}} is not valid. Use Allow or Deny.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The effect {{effect}} is not valid. Use Allow or Deny."
```

Risoluzione dell'errore

Aggiorna l'elemento `Effect` per utilizzare un effetto valido. I valori validi di `Effect` sono **`Allow`** e **`Deny`**.

Termini correlati

- Elemento Effetto
- Panoramica delle policy JSON

Errore: chiave di condizione globale non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid global condition key: The condition key {{key}} does not exist. Use a valid
 condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} does not exist. Use a valid condition
 key."
```

Risoluzione dell'errore

Aggiorna la chiave di condizione nella coppia chiave-valore della condizione per utilizzare una chiave di condizione globale supportata.

Le chiavi di condizione globali sono chiavi di condizione con un `aws:` prefisso. AWS i servizi possono supportare chiavi di condizione globali o fornire chiavi specifiche del servizio che includono il relativo prefisso di servizio. Ad esempio, le chiavi di condizione IAM includono il prefisso `iam:`. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e scegli il servizio di cui desideri visualizzare le chiavi.

Termini correlati

- Chiavi della condizione globale

Errore: partizione non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid partition: The resource ARN for the service {{service}} does not support the
 partition {{partition}}. Use the supported values: {{partitions}}
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN for the service {{service}} does not support the
 partition {{partition}}. Use the supported values: {{partitions}}"
```

Risoluzione dell'errore

Aggiorna l'ARN della risorsa per includere una partizione supportata. Se hai incluso una partizione supportata, il servizio o la risorsa potrebbe non supportare la partizione inclusa.

Una partizione è un gruppo di regioni. AWS Ogni AWS account è limitato a una partizione. Nelle regioni classiche, utilizza la partizione `aws`. Nelle regioni della Cina, utilizza `aws-cn`.

Termini correlati

- Amazon Resource Name (ARN) - Partizioni

Errore: elemento della policy non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid policy element: The policy element {{element}} is not valid.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy element {{element}} is not valid."
```

Risoluzione dell'errore

Aggiorna la policy per includere solo gli elementi della policy JSON supportati.

Termini correlati

- Elementi delle policy JSON

Errore: formato principale non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid principal format: The Principal element contents are not valid. Specify a key-
value pair in the Principal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element contents are not valid. Specify a key-value
 pair in the Principal element."
```

Risoluzione dell'errore

Aggiorna il principale per utilizzare un formato di coppia chiave-valore supportato.

Puoi specificare un principale in una policy basata sulle risorse, ma non in una policy basata sulle identità.

Ad esempio, per definire l'accesso per tutti gli utenti di un AWS account, utilizza il seguente principio nella tua politica:

```
"Principal": { "AWS": "123456789012" }
```

Termini correlati

- Elementi delle policy JSON: principale
- Policy basate sulle identità e policy basate su risorse

Errore: chiave principale non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid principal key: The principal key {{principal-key}} is not valid.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The principal key {{principal-key}} is not valid."
```

Risoluzione dell'errore

Aggiorna la chiave nella coppia chiave-valore del principale per utilizzare una chiave principale supportata. Le chiavi principali supportate sono le seguenti:

- AWS
- CanonicalUser
- Federato
- Servizio

Termini correlati

- Elemento principale

Errore: regione non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid Region: The Region {{region}} is not valid. Update the condition value to a
 suported Region.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Region {{region}} is not valid. Update the condition value to a
 suported Region."
```

Risoluzione dell'errore

Aggiorna il valore della coppia chiave-valore della condizione per includere una regione supportata. Per una tabella dei AWS servizi supportati in ogni regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse

- Nomi e codici delle regioni

Errore: servizio non valido

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Invalid service: The service {{service}} does not exist. Use a valid service name.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} does not exist. Use a valid service name."
```

Risoluzione dell'errore

Il prefisso del servizio nell'operazione o nella chiave di condizione deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Servizi noti e relative operazioni, risorse e chiavi di condizione

Errore: chiave di condizione del servizio non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid service condition key: The condition key {{key}} does not exist in the service
 {{service}}. Use a valid condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{key}} does not exist in the service {{service}}.
 Use a valid condition key."
```

Risoluzione dell'errore

Aggiorna la chiave nella coppia chiave-valore della condizione per utilizzare una chiave di condizione nota per il servizio. Le chiavi di condizione globali sono chiavi di condizione con un prefisso `aws`. I servizi AWS possono fornire chiavi specifiche del servizio che includono il prefisso del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi.

Termini correlati

- Chiavi della condizione globale
- Servizi noti e relative operazioni, risorse e chiavi di condizione

Errore: servizio non valido nell'operazione

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid service in action: The service {{service}} specified in the action does not
 exist. Did you mean {{service2}}?
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service {{service}} specified in the action does not exist. Did
 you mean {{service2}}?"
```

Risoluzione dell'errore

Il prefisso del servizio nell'operazione deve corrispondere alle specifiche (incluso il formato maiuscolo/minuscolo) per i prefissi del servizio. Per visualizzare il prefisso di un servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi. Scegli il nome del servizio e individua il suo prefisso nella prima frase.

Termini correlati

- Elemento dell'operazione
- Servizi noti e relative operazioni

Errore: variabile non valida per l'operatore

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid variable for operator: Policy variables can only be used with String and ARN
 operators.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Policy variables can only be used with String and ARN operators."
```

Risoluzione dell'errore

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`. Le condizioni supportano le variabili quando si utilizzano operatori stringa o operatori ARN. Gli operatori stringa includono `StringEquals`, `StringLike` e `StringNotLike`. Gli operatori ARN includono `ArnEquals` e `ArnLike`. Non è possibile utilizzare una variabile della policy con altri operatori, ad esempio Numeric, Date, Boolean, Binary, IP Address o Null.

Termini correlati

- Utilizzo delle variabili delle policy nell'elemento Condizione
- Elemento condizione

Errore: versione non valida

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid version: The version ${version} is not valid. Use one of the following
 versions: ${versions}
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The version ${version} is not valid. Use one of the following
 versions: ${versions}"
```

Risoluzione dell'errore

L'elemento `Version` policy specifica le regole di sintassi del linguaggio AWS utilizzate per elaborare una policy. Per utilizzare tutte le funzionalità della policy disponibili, includi il seguente elemento `Version` prima dell'elemento `Statement` in tutte le policy.

```
"Version": "2012-10-17"
```

Termini correlati

- Elemento della versione

Errore: errore di sintassi JSON

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Json syntax error: Fix the JSON syntax error at index {{index}} line {{line}} column
 {{column}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Fix the JSON syntax error at index {{index}} line {{line}} column
 {{column}}."
```

Risoluzione dell'errore

La policy include un errore di sintassi. Controlla la sintassi JSON.

Termini correlati

- Validatore JSON
- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: errore di sintassi JSON

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Json syntax error: Fix the JSON syntax error.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Fix the JSON syntax error."
```

Risoluzione dell'errore

La policy include un errore di sintassi. Controlla la sintassi JSON.

Termini correlati

- Validatore JSON
- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: operazione mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing action: Add an Action or NotAction element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add an Action or NotAction element to the policy statement."
```

Risoluzione dell'errore

AWS Le politiche JSON devono includere un elemento `Action` or`NotAction`.

Termini correlati

- Elemento dell'operazione
- NotAction elemento
- Panoramica delle policy JSON

Errore: campo ARN mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing ARN field: Resource ARNs must include at least {{fields}} fields in the
 following structure: arn:partition:service:region:account:resource
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Resource ARNs must include at least {{fields}} fields in the
 following structure: arn:partition:service:region:account:resource"
```

Risoluzione dell'errore

Tutti i campi nell'ARN della risorsa devono corrispondere alle specifiche per i tipi di risorse noti. Per visualizzare il formato ARN previsto per un servizio, vedere Azioni, risorse e chiavi di condizione per i AWS servizi. Sceglie il nome del servizio per visualizzarne i tipi di risorse e i formati ARN.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- AWS risorse di servizio con formati ARN

Errore: regione ARN mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing ARN Region: Add a Region to the {{service}} resource ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Region to the {{service}} resource ARN."
```

Risoluzione dell'errore

Gli ARN delle risorse per la maggior parte dei AWS servizi richiedono la specificazione di una regione. Per una tabella dei AWS servizi supportati in ogni regione, consulta la tabella delle regioni.

Termini correlati

- Risorse relative alle policy
- ARN delle risorse
- Nomi e codici delle regioni

Errore: effetto mancante

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Missing effect: Add an Effect element to the policy statement with a value of Allow or
 Deny.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add an Effect element to the policy statement with a value of Allow
 or Deny."
```

Risoluzione dell'errore

AWS Le politiche JSON devono includere un `Effect` elemento con un valore di **`Allow`** e. **`Deny`**

Termini correlati

- Elemento Effetto
- Panoramica delle policy JSON

Errore: principale mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing principal: Add a Principal element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Principal element to the policy statement."
```

Risoluzione dell'errore

Le policy basate sulle risorse devono includere un elemento `Principal`.

Ad esempio, per definire l'accesso per tutti gli utenti di un AWS account, utilizza il seguente principio nella tua politica:

```
"Principal": { "AWS": "123456789012" }
```

Termini correlati

- Elemento principale
- Policy basate sulle identità e policy basate su risorse

Errore: qualificatore mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing qualifier: The request context key ${key} has multiple values. Use the
 ForAllValues or ForAnyValue condition key qualifiers in your policy.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The request context key ${key} has multiple values. Use the
 ForAllValues or ForAnyValue condition key qualifiers in your policy."
```

Risoluzione dell'errore

Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali ("uguale a", "minore di" e così via) per confrontare le chiavi e i valori della policy rispetto alle chiavi e ai valori del contesto della richiesta. Per le richieste che includono più valori per una singola chiave, è necessario racchiudere le condizioni tra parentesi come un array ("Key2":["Value2A", "Value2B"]). È inoltre necessario utilizzare gli operatori su set `ForAllValues` o `ForAnyValue` con l'operatori di condizione `StringLike`. Questi qualificatori aggiungono funzionalità di operazione set all'operatore della condizione, in modo che sia possibile testare più valori di richieste con più valori di condizione.

Termini correlati

- Chiavi di contesto multivalore
- Elemento condizione

AWS politiche gestite con questo errore

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono un qualificatore mancante per le chiavi di condizione nelle relative dichiarazioni politiche. Quando si utilizza la politica AWS gestita come riferimento per creare una politica gestita dai clienti, si AWS consiglia di aggiungere i qualificatori chiave `ForAllValues` o `ForAnyValue` condizionali all'elemento. `Condition`

- AWSGlueConsoleSageMakerNotebookFullAccess

Errore: risorsa mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing resource: Add a Resource or NotResource element to the policy statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a Resource or NotResource element to the policy statement."
```

Risoluzione dell'errore

Tutte le politiche, ad eccezione delle politiche di trust dei ruoli, devono includere un `NotResource` elemento `Resource` or.

Termini correlati

- Elemento risorsa
- NotResource elemento
- Policy basate sulle identità e policy basate su risorse
- Panoramica delle policy JSON

Errore: istruzione mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing statement: Add a statement to the policy
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a statement to the policy"
```

Risoluzione dell'errore

Una policy JSON deve includere un'istruzione.

Termini correlati

- Elementi delle policy JSON

Errore: null con if esiste

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Null with if exists: The Null condition operator cannot be used with the IfExists
 suffix. Update the operator or the suffix.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Null condition operator cannot be used with the IfExists suffix.
 Update the operator or the suffix."
```

Risoluzione dell'errore

È possibile aggiungere `IfExists` alla fine di qualsiasi nome dell'operatore di condizione ad eccezione dell'operatore di condizione `Null`. Utilizza un operatore di condizione `Null` per verificare se una chiave di condizione è presente al momento dell'autorizzazione. Utilizza `...ifExists` per dichiarare che "Se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere elaborata come specificato nella policy. Se la chiave non è presente, l'elemento della condizione viene valutato come "true".

Termini correlati

- ... IfExists operatori di condizionamento
- Operatore di condizione null
- Elemento condizione

Errore: carattere jolly dell'operazione con errore di sintassi SCP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error action wildcard: SCP actions can include wildcards (*) only at the end
 of a string. Update {{action}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCP actions can include wildcards (*) only at the end of a string.
 Update {{action}}."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) supportano la specificazione di valori negli `Action` elementi or. `NotAction` Tuttavia, questi valori possono includere caratteri jolly (*) solo alla fine della stringa. Ciò significa che puoi specificare `iam:Get*` ma non `iam:*role`.

Per specificare più azioni, si AWS consiglia di elencarle singolarmente.

Termini correlati

- Azioni ed NotAction elementi SCP
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio
- Elementi delle policy JSON IAM: Action

Errore: condizione di autorizzazione con errore di sintassi SCP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error allow condition: SCPs do not support the Condition element with effect
 Allow. Update the element Condition or the effect.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support the Condition element with effect Allow. Update
 the element Condition or the effect."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) supportano la specificazione di valori nell'`Condition`elemento solo quando si utilizza. `"Effect": "Deny"`

Per consentire solo una singola operazione, è possibile negare l'accesso a tutto tranne la condizione specificata utilizzando la versione `...NotEquals` di un operatore di condizione. Questo rifiuta il confronto fatto dall'operatore.

Termini correlati

- Elemento di condizione SCP
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio
- Politica di esempio: nega l'accesso AWS in base alla regione richiesta
- Elementi della policy JSON IAM: operatori di condizione
- Elementi della policy JSON IAM: Condition

Errore: errore di sintassi SCP consentito NotAction

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error allow NotAction: SCPs do not support NotAction with effect Allow.
 Update the element NotAction or the effect.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support NotAction with effect Allow. Update the element
 NotAction or the effect."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) non supportano l'utilizzo dell'`NotAction`elemento con. `"Effect": "Allow"`

È necessario riscrivere la logica per consentire una lista di operazioni o per rifiutare tutte le operazioni che non sono nell'elenco.

Termini correlati

- Azioni ed elementi SCP NotAction
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio
- Elementi delle policy JSON IAM: Action

Errore: risorsa di autorizzazione con errore della sintassi SCP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error allow resource: SCPs do not support Resource with effect Allow. Update
 the element Resource or the effect.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support Resource with effect Allow. Update the element
 Resource or the effect."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) supportano la specificazione di valori nell'`Resource`elemento solo quando si utilizza. `"Effect": "Deny"`

È necessario riscrivere la logica per consentire tutte le risorse o rifiutare tutte le risorse elencate.

Termini correlati

- Elemento risorsa SCP
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio

- Elementi delle policy JSON IAM: Resource

Errore: errore di sintassi SCP NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error NotResource: SCPs do not support the NotResource element. Update the
 policy to use Resource instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support the NotResource element. Update the policy to
 use Resource instead."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) non supportano l'`NotResource`elemento.

È necessario riscrivere la logica per consentire tutte le risorse o rifiutare tutte le risorse elencate.

Termini correlati

- Elemento risorsa SCP
- Valutazione SCP
- AWS Organizations politiche di controllo del servizio
- Elementi delle policy JSON IAM: Resource

Errore: principale dell'errore di sintassi SCP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
SCP syntax error principal: SCPs do not support specifying principals. Remove the
 Principal or NotPrincipal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "SCPs do not support specifying principals. Remove the Principal or
 NotPrincipal element."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) non supportano gli elementi `Principal` or`NotPrincipal`.

Puoi specificare l'Amazon Resource Name (ARN) utilizzando la chiave di condizione globale `aws:PrincipalArn` nell'elemento `Condition`.

Termini correlati

- Sintassi delle SCP
- Chiavi di condizione globali per i principali

Errore: sid univoci richiesti

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unique Sids required: Duplicate statement IDs are not supported for this policy type.
 Update the Sid value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Duplicate statement IDs are not supported for this policy type.
 Update the Sid value."
```

Risoluzione dell'errore

Per alcuni tipi di policy , gli ID delle istruzioni devono essere univoci. L'elemento `Sid` (ID istruzione) consente di immettere un identificatore opzionale fornito per l'istruzione della policy. Puoi assegnare un valore ID di istruzione a ogni istruzione in una matrice di istruzioni utilizzando l'elemento `SID`. Nei servizi che consentono di specificare un elemento ID, ad esempio SQS e SNS, il valore `Sid` è semplicemente un ID secondario dell'ID del documento di policy. Ad esempio, in IAM il valore `Sid` deve essere univoco all'interno di una policy JSON.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Errore: operazione non supportata nella policy

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action in policy: The action {{action}} is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcune operazioni non sono supportate nell'elemento `Action` nella policy basata su risorse collegato a un tipo di risorsa diverso. Ad esempio, AWS Key Management Service le azioni non sono supportate nelle policy dei bucket di Amazon S3. Specificare un'operazione supportata dal tipo di risorsa collegato alla policy basata su risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: combinazione di elementi non supportata

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported element combination: The policy elements ${element1} and ${element2} can
 not be used in the same statement. Remove one of these elements.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy elements ${element1} and ${element2} can not be used in
 the same statement. Remove one of these elements."
```

Risoluzione dell'errore

Alcune combinazioni di elementi delle policy JSON non possono essere utilizzate insieme. Ad esempio, non è possibile utilizzare `Action` e `NotAction` nella stessa dichiarazione di policy. Altre coppie che si escludono reciprocamente sono `Principal/NotPrincipal` e `Resource/NotResource`.

Termini correlati

- Documentazione di riferimento degli elementi delle policy JSON IAM
- Panoramica delle policy JSON

Errore: chiave di condizione globale non supportata

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported global condition key: The condition key aws:ARN is not supported. Use
 aws:PrincipalArn or aws:SourceArn instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key aws:ARN is not supported. Use aws:PrincipalArn or
 aws:SourceArn instead."
```

Risoluzione dell'errore

AWS non supporta l'utilizzo della chiave di condizione globale specificata. A seconda del tuo caso d'uso, puoi utilizzare le chiavi di condizione globali `aws:PrincipalArn` o `aws:SourceArn`. Ad esempio, invece di `aws:ARN` utilizza `aws:PrincipalArn` per confrontare l'Amazon Resource Name (ARN del principale che ha effettuato la richiesta con l'ARN specificato nella policy. In alternativa, utilizza la chiave `aws:SourceArn` global condition per confrontare l'Amazon Resource Name (ARN) della risorsa che effettua una service-to-service richiesta con l'ARN specificato nella policy.

Termini correlati

- AWS chiavi di contesto della condizione globale

Errore: principale non supportato

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported principal: The policy type ${policy_type} does not support the Principal
 element. Remove the Principal element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy type ${policy_type} does not support the Principal
 element. Remove the Principal element."
```

Risoluzione dell'errore

L'elemento `Principal` specifica il principale a cui è consentito o rifiutato l'accesso a una risorsa. Non è possibile utilizzare l'elemento `Principal` in una policy basata sull'identità IAM. Puoi usarlo nelle policy di attendibilità per i ruoli IAM e nelle policy basate sulle risorse. Le policy basate su risorse sono policy che vengono incorporate direttamente in una risorsa. Ad esempio, puoi incorporare le policy in un bucket Amazon S3 o AWS in una chiave KMS.

Termini correlati

- AWS Elementi delle policy JSON: principale
- Accesso alle risorse multi-account in IAM

Errore: ARN della risorsa non supportata nella policy

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported resource ARN in policy: The resource ARN is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcuni ARN delle risorse non sono supportati nell'elemento `Resource` della policy basata sulle risorse quando la policy è collegata a un tipo di risorsa diverso. Ad esempio, gli AWS KMS ARN non

sono supportati nell'`Resource`elemento per le policy dei bucket di Amazon S3. Specificare un ARN della risorsa supportato da un tipo di risorsa collegato alla policy basata sulle risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: sid non supportato

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported Sid: Update the characters in the Sid element to use one of the following
 character types: [a-z, A-Z, 0-9]
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Update the characters in the Sid element to use one of the following
 character types: [a-z, A-Z, 0-9]"
```

Risoluzione dell'errore

L'elemento `Sid` supporta lettere maiuscole, lettere minuscole e numeri.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Errore: carattere jolly non supportato nel principale

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported wildcard in principal: Wildcards (*, ?) are not supported with the
 principal key {{principal_key}}. Replace the wildcard with a valid principal value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Wildcards (*, ?) are not supported with the principal key
 {{principal_key}}. Replace the wildcard with a valid principal value."
```

Risoluzione dell'errore

La struttura dell'elemento `Principal` supporta l'utilizzo di una coppia chiave-valore. Il valore del principale specificato nella policy include un carattere jolly (*). Non è possibile includere un carattere jolly con la chiave del principale specificata. Ad esempio, quando specifichi gli utenti in un elemento `Principal`, non è possibile utilizzare un carattere jolly che indica "tutti gli utenti". Assegna un nome a uno o più utenti specifici. Allo stesso modo, quando si specifica una sessione con assunzione di ruolo, non è possibile utilizzare un carattere jolly (*) per indicare "tutte le sessioni". È necessario assegnare un nome a una sessione specifica. Non è possibile utilizzare un carattere jolly per associare parte di un nome o di un ARN.

Per risolvere questo risultato, rimuovi il carattere jolly e fornisci un principale più specifico.

Termini correlati

- AWS Elementi delle policy JSON: principale

Errore: parentesi mancante nella variabile

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing brace in variable: The policy variable is missing a closing curly brace. Add }
 after the variable text.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy variable is missing a closing curly brace. Add } after
 the variable text."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy.

Per risolvere questo risultato, aggiungi la parentesi graffa mancante per assicurarti che sia presente il set completo di parentesi graffe di apertura e chiusura.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: virgoletta mancante nella variabile

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing quote in variable: The policy variable default value must begin and end with a
 single quote. Add the missing quote.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy variable default value must begin and end with a single
 quote. Add the missing quote."
```

Risoluzione dell'errore

Quando aggiungi una variabile alla policy, puoi specificare un valore di default per la variabile. Se una variabile non è presente, AWS utilizza il testo predefinito fornito dall'utente.

Per aggiungere un valore di default a una variabile, racchiudi il valore di default tra virgolette singole (`' '`) e separa il testo della variabile e il valore di default con una virgola e uno spazio (`, `).

Ad esempio, se un principale è contrassegnato con `team=yellow`, possono accedere al bucket Amazon S3 `DOC-EXAMPLE-BUCKET` con il nome`DOC-EXAMPLE-BUCKET-yellow`. Una policy con questa risorsa potrebbe consentire ai membri del team di accedere alle proprie risorse, ma non a quelle di altri team. Per gli utenti senza tag dei team, puoi impostare un valore di default di`company-wide`. Questi utenti possono accedere solo al bucket `DOC-EXAMPLE-BUCKET-company-wide`, dove possono visualizzare informazioni generali, come le istruzioni per entrare a far parte di un team.

```
"Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET-${aws:PrincipalTag/team, 'company-wide'}"
```

Termini correlati

- Elementi delle policy IAM: variabili

Errore: spazio non supportato nella variabile

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported space in variable: A space is not supported within the policy variable
 text. Remove the space.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "A space is not supported within the policy variable text. Remove the
 space."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy. Sebbene sia possibile includere uno spazio quando si specifica una variabile di default, non è possibile includere uno spazio nel nome della variabile.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile vuota

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty variable: Empty policy variable. Remove the ${ } variable structure or provide a
 variable within the structure.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Empty policy variable. Remove the ${ } variable structure or provide
 a variable within the structure."
```

Risoluzione dell'errore

La struttura delle variabili delle policy supporta l'utilizzo di un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ }, includi il nome del valore ricavato dalla richiesta da utilizzare nella policy.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile non supportata nell'elemento

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Variable unsupported in element: Policy variables are supported in the Resource and
 Condition elements. Remove the policy variable {{variable}} from this element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Policy variables are supported in the Resource and Condition
 elements. Remove the policy variable {{variable}} from this element."
```

Risoluzione dell'errore

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`.

Termini correlati

- Elementi delle policy IAM: variabili

Errore: variabile non supportata nella versione

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Variable unsupported in version: To include variables in your policy, use the policy
 version 2012-10-17 or later.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "To include variables in your policy, use the policy version
 2012-10-17 or later."
```

Risoluzione dell'errore

Per usare le variabili di policy, è necessario che l'elemento `Version` sia incluso in una istruzione e impostato su una versione che supporti le variabili di policy. Le variabili sono state introdotte a partire dalla versione `2012-10-17`. Le versioni precedenti del linguaggio di policy non supportano le variabili. Se non imposti la `Version` su `2012-10-17` o una versione successiva, le variabili come `${aws:username}` nella policy vengono trattate come stringhe letterali.

Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Una versione della policy viene creata quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita.

Termini correlati

- Elementi delle policy IAM: variabili
- Elementi delle policy JSON IAM: Version

Errore: indirizzo IP privato

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Private IP address: aws:SourceIp works only for public IP address ranges. The values
 for condition key aws:SourceIp include only private IP addresses and will not have the
 desired effect. Update the value to include only public IP addresses.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "aws:SourceIp works only for public IP address ranges. The values
 for condition key aws:SourceIp include only private IP addresses and will not have the
 desired effect. Update the value to include only public IP addresses."
```

Risoluzione dell'errore

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici. Questo errore viene visualizzato quando la policy consente solo indirizzi IP privati. In questo caso, la condizione non corrisponderà mai.

- aws: chiave di condizione SourceIp globale
- Elementi della policy JSON IAM: Condition

Errore: privato NotIpAddress

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Private NotIpAddress: The values for condition key aws:SourceIp include only private
 IP addresses and has no effect. aws:SourceIp works only for public IP address ranges.
 Update the value to include only public IP addresses.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include only private IP
 addresses and has no effect. aws:SourceIp works only for public IP address ranges.
 Update the value to include only public IP addresses."
```

Risoluzione dell'errore

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici. Questo errore viene visualizzato quando si utilizza l'operatore di condizione `NotIpAddress` e sono riportati solo gli indirizzi IP privati. In questo caso, la condizione corrispondere sempre ed è inefficace.

- aws: chiave di condizione SourceIp globale
- Elementi della policy JSON IAM: Condition

Errore: la dimensione della policy supera la quota della SCP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds SCP quota: The {{policySize}} characters in the service control
 policy (SCP) exceed the {{policySizeQuota}} character maximum for SCPs. We recommend
 that you use multiple granular policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the service control policy (SCP)
 exceed the {{policySizeQuota}} character maximum for SCPs. We recommend that you use
 multiple granular policies."
```

Risoluzione dell'errore

AWS Organizations le politiche di controllo dei servizi (SCP) supportano la specificazione di valori negli `Action` elementi or. `NotAction` Tuttavia, questi valori possono includere caratteri jolly (*) solo alla fine della stringa. Ciò significa che puoi specificare `iam:Get*` ma non `iam:*role`.

Per specificare più azioni, si AWS consiglia di elencarle singolarmente.

Termini correlati

- Quote per le organizzazioni AWS
- AWS Organizations politiche di controllo del servizio

Errore: formato principale del servizio non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid service principal format: The service principal does not match the expected
 format. Use the format {{expectedFormat}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The service principal does not match the expected format. Use the
 format {{expectedFormat}}."
```

Risoluzione dell'errore

Il valore nella coppia chiave-valore della condizione deve corrispondere a un formato di principale di servizio definito.

Un'entità servizio è un identificatore che viene utilizzato per concedere autorizzazioni a un servizio. Puoi specificare un principale di servizio nell'elemento `Principal` o come valore per alcune chiavi di condizione globali e chiavi specifiche del servizio. Il principale del servizio è definito da ciascun servizio.

L'identificatore di un principale del servizio include il nome del servizio ed è solitamente nel formato seguente con tutte le lettere minuscole:

service-name.amazonaws.com

Alcune chiavi specifiche del servizio possono utilizzare un formato diverso per i principali di servizio. Ad esempio, la chiave di condizione `kms:ViaService` richiede il seguente formato per i principali del servizio con tutte le lettere minuscole:

service-name.AWS_region`.amazonaws.com`

Termini correlati

- Principali del servizio
- AWS chiavi di condizione globali
- Chiave di condizione `kms:ViaService`

Errore: chiave di tag mancante nella condizione

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing tag key in condition: The condition key {{conditionKeyName}} must include a tag
 key to control access based on tags. Use the format {{conditionKeyName}}tag-key and
 specify a key name for tag-key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{conditionKeyName}} must include a tag key to
 control access based on tags. Use the format {{conditionKeyName}}tag-key and specify a
 key name for tag-key."
```

Risoluzione dell'errore

Per controllare gli accessi in base ai tag, devi fornire informazioni sui tag nell'elemento condizione di una policy.

Ad esempio, per controllare l'accesso alle AWS risorse, si include la chiave di `aws:ResourceTag` condizione. Questa chiave richiede il formato `aws:ResourceTag/`*`tag-key`*. Per specificare la chiave di tag `owner` e il valore del tag `JaneDoe` In una condizione, utilizza il seguente formato:

```
"Condition": {
    "StringEquals": {"aws:ResourceTag/owner": "JaneDoe"}
```

```
}
```

Termini correlati

- Controllo degli accessi tramite tag
- Condizioni
- Chiavi della condizione globale
- AWS chiavi delle condizioni di servizio

Errore: formato vpc non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid vpc format: The VPC identifier in the condition key value is not valid. Use the
 prefix 'vpc-' followed by 8 or 17 alphanumeric characters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The VPC identifier in the condition key value is not valid. Use the
 prefix 'vpc-' followed by 8 or 17 alphanumeric characters."
```

Risoluzione dell'errore

La chiave di condizione `aws:SourceVpc` deve utilizzare il prefisso `vpc-` seguito da 8 o 17 caratteri alfanumerici, ad esempio `vpc-11223344556677889` o `vpc-12345678`.

Termini correlati

- AWS chiavi di condizione globali: aws: SourceVpc

Errore: formato vpce non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid vpce format: The VPCE identifier in the condition key value is not valid.  Use
 the prefix 'vpce-' followed by 8 or 17 alphanumeric characters.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The VPCE identifier in the condition key value is not valid.  Use
 the prefix 'vpce-' followed by 8 or 17 alphanumeric characters."
```

Risoluzione dell'errore

La chiave di condizione `aws:SourceVpce` deve utilizzare il prefisso `vpce-` seguito da 8 o 17 caratteri alfanumerici, ad esempio `vpce-11223344556677889` o `vpce-12345678`.

Termini correlati

- AWS chiavi di condizione globali: aws: SourceVpce

Errore: principale federato non supportato

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Federated principal not supported: The policy type does not support a federated
 identity provider in the principal element. Use a supported principal.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The policy type does not support a federated identity provider in
 the principal element. Use a supported principal."
```

Risoluzione dell'errore

L'elemento `Principal` utilizza i principali federati per le policy di attendibilità collegate ai ruoli IAM per fornire l'accesso tramite la federazione delle identità. Le policy di identità e altre policy basate sulle risorse non supportano un provider di identità federato nell'elemento `Principal`. Ad esempio, non puoi utilizzare un principale SAML in una policy bucket Amazon S3. Modifica l'elemento `Principal` con un tipo di principale supportato.

Termini correlati

- Creazione di un ruolo per la federazione delle identità
- Elementi delle policy JSON: principale

Errore: operazione non supportata per la chiave di

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action for condition key: The following actions: {{actions}} are not
 supported by the condition key {{key}}. The condition will not be evaluated for these
 actions. We recommend that you move these actions to a different statement without
 this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The following actions: {{actions}} are not supported by the
 condition key {{key}}. The condition will not be evaluated for these actions. We
 recommend that you move these actions to a different statement without this condition
 key."
```

Risoluzione dell'errore

Assicurati che la chiave di condizione sia nell'elemento `Condition` della dichiarazione di policy si applichi a tutte le operazioni nell'elemento `Action`. Per garantire che le operazioni specificate siano effettivamente consentite o rifiutate dalla policy, è necessario spostare le operazioni non supportate in una dichiarazione diversa senza la chiave di condizione.

> **Note**
>
> Se l'elemento `Action` ha operazioni con caratteri jolly, Sistema di analisi degli accessi AWS IAM non le valuta per questo errore.

Termini correlati

- Elementi delle policy JSON: Action

Errore: operazione non supportata nella policy

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported action in policy: The action {{action}} is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The action {{action}} is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcune operazioni non sono supportate nell'elemento `Action` nella policy basata su risorse collegato a un tipo di risorsa diverso. Ad esempio, AWS Key Management Service le azioni non sono supportate nelle policy dei bucket di Amazon S3. Specificare un'operazione supportata dal tipo di risorsa collegato alla policy basata su risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: ARN della risorsa non supportata nella policy

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported resource ARN in policy: The resource ARN is not supported for the resource-
based policy attached to the resource type {{resourceType}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The resource ARN is not supported for the resource-based policy
 attached to the resource type {{resourceType}}."
```

Risoluzione dell'errore

Alcuni ARN delle risorse non sono supportati nell'elemento `Resource` della policy basata sulle risorse quando la policy è collegata a un tipo di risorsa diverso. Ad esempio, gli AWS KMS ARN non sono supportati nell'`Resource`elemento per le policy dei bucket di Amazon S3. Specificare un ARN della risorsa supportato da un tipo di risorsa collegato alla policy basata sulle risorse.

Termini correlati

- Elementi delle policy JSON: Action

Errore: chiave di condizione non supportata per il principale del servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unsupported condition key for service principal: The following condition keys are not
 supported when used with the service principal: {{conditionKeys}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The following condition keys are not supported when used with the
 service principal: {{conditionKeys}}."
```

Risoluzione dell'errore

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un service principal, che è un identificatore del servizio. Non è possibile utilizzare alcune chiavi di condizione con determinati principali del servizio. Ad esempio, non puoi utilizzare la chiave di condizione `aws:PrincipalOrgID` con il principale del servizio `cloudfront.amazonaws.com`. È necessario rimuovere le chiavi di condizione che non si applicano al principale del servizio nell'elemento `Principal`.

Termini correlati

- Principali del servizio
- Elementi delle policy JSON: principale

Errore: errore di sintassi notprincipal della policy di attendibilità del ruolo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy syntax error notprincipal: Role trust policies do not support
 NotPrincipal. Update the policy to use a Principal element instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Role trust policies do not support NotPrincipal. Update the policy
 to use a Principal element instead."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Le politiche di attendibilità dei ruoli non supportano `NotPrincipal`. Aggiornare la policy per utilizzare un elemento `Principal`.

Termini correlati

- Elementi delle policy JSON: principale
- Elementi della policy JSON: NotPrincipal

Errore: carattere jolly della policy di attendibilità del ruolo non supportato nel principale

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy unsupported wildcard in principal: "Principal:" "*" is not supported
 in the principal element of a role trust policy. Replace the wildcard with a valid
 principal value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": ""Principal:" "*" is not supported in the principal element of a role
 trust policy. Replace the wildcard with a valid principal value."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. L'elemento `Principal` della policy di attendibilità del ruolo non supporta `"Principal:" "*"`. Sostituisci il jolly con un valore principale valido.

Termini correlati

- Elementi delle policy JSON: principale

Error: errore di sintassi resource della policy di attendibilità del ruolo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Role trust policy syntax error resource: Role trust policies apply to the role
 that they are attached to. You cannot specify a resource. Remove the Resource or
 NotResource element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Role trust policies apply to the role that they are attached to. You
 cannot specify a resource. Remove the Resource or NotResource element."
```

Risoluzione dell'errore

Una policy di attendibilità del ruolo è una policy basata sulle risorse collegata a un ruolo IAM. Le policy di attendibilità definiscono quali entità principali (account, utenti, ruoli e utenti federati) possono assumere il ruolo. Le politiche di attendibilità del ruolo si applicano al ruolo a cui sono associate. Non puoi specificare un elemento `Resource` o `NotResource` in una policy di attendibilità del ruolo. Rimozione dell'elemento `Resource` o `NotResource`.

- Elementi delle policy JSON: Resource
- Elementi della policy JSON: NotResource

Errore: il tipo non corrisponde all'intervallo IP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch IP range: The condition operator {{operator}} is used with an invalid IP
 range value. Specify the IP range in standard CIDR format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition operator {{operator}} is used with an invalid IP range
 value. Specify the IP range in standard CIDR format."
```

Risoluzione dell'errore

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione dell'indirizzo IP, in un formato CIDR.

Termini correlati

- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: operatori di condizione

Errore: operazione mancante per la chiave di condizione

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing action for condition key: The {{actionName}} action must be in the action block
 to allow setting values for the condition key {{keyName}}. Add {{actionName}} to the
 action block.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{actionName}} action must be in the action block to allow
 setting values for the condition key {{keyName}}. Add {{actionName}} to the action
 block."
```

Risoluzione dell'errore

La chiave della condizione nell'elemento `Condition` della dichiarazione di policy non viene valutata a meno che l'operazione specificata non sia inclusa nell'elemento `Action`. Per garantire che le chiavi di condizione specificate siano effettivamente consentite o rifiutate dalla policy, aggiungi l'operazione all'elemento `Action`.

Termini correlati

- Elementi delle policy JSON: Action

Errore: sintassi del principale federato non valida nella policy di attendibilità dei ruoli

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid federated principal syntax in role trust policy: The principal value specifies
 a federated principal that does not match the expected format. Update the federated
 principal to a domain name or a SAML metadata ARN.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The principal value specifies a federated principal that does not
 match the expected format. Update the federated principal to a domain name or a SAML
 metadata ARN."
```

Risoluzione dell'errore

Il valore principale specifica un pricipale federato che non corrisponde al formato previsto. Aggiorna il formato del principale federato con un nome di dominio valido o un ARN di metadati SAML.

Termini correlati

- Utenti federati e ruoli

Errore: operazione non corrispondente per il principale

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Mismatched action for principal: The {{actionName}} action is invalid with the
 following principal(s): {{principalNames}}. Use a SAML provider principal with
 the sts:AssumeRoleWithSAML action or use an OIDC provider principal with the
 sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if you use
 either of the two options.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{actionName}} action is invalid with the following
 principal(s): {{principalNames}}. Use a SAML provider principal with the
 sts:AssumeRoleWithSAML action or use an OIDC provider principal with the
 sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if you use
 either of the two options."
```

Risoluzione dell'errore

L'operazione specificata nell'elemento `Action` della dichiarazione di policy non è valida con il principale specificato nell'elemento `Principal`. Ad esempio, non puoi utilizzare un principale

provider SAML con l'operazione `sts:AssumeRoleWithWebIdentity`. È necessario utilizzare un provider principale SAML con l'operazione `sts:AssumeRoleWithSAML` oppure utilizzare un principale del fornitore OIDC con l'operazione `sts:AssumeRoleWithWebIdentity`.

Termini correlati

- AssumeRoleWithSAML
- AssumeRoleWithWebIdentity

Errore: operazione mancante per la policy di attendibilità dei ruoli ovunque

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing action for roles anywhere trust policy: The rolesanywhere.amazonaws.com service
 principal requires the sts:AssumeRole, sts:SetSourceIdentity, and sts:TagSession
 permissions to assume a role. Add the missing permissions to the policy.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The rolesanywhere.amazonaws.com service principal requires the
 sts:AssumeRole, sts:SetSourceIdentity, and sts:TagSession permissions to assume a
 role. Add the missing permissions to the policy."
```

Risoluzione dell'errore

Affinché IAM Roles Anywhere sia in grado di assumere un ruolo e fornire credenziali AWS temporanee, il ruolo deve considerare attendibile il principale del servizio IAM Roles Anywhere. Il principale del servizio IAM Roles Anywhere richiede le autorizzazioni `sts:AssumeRole`, `sts:SetSourceIdentity` e `sts:TagSession` per assumere un ruolo. Se manca una delle autorizzazioni, va aggiunta alla policy.

Termini correlati

- Modello di fiducia in AWS Identity and Access Management Roles Anywhere

Avviso generale: crea SLR con NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with NotResource: Using the iam:CreateServiceLinkedRole action with
 NotResource can allow creation of unintended service-linked roles for multiple
 resources. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:CreateServiceLinkedRole action with NotResource can
 allow creation of unintended service-linked roles for multiple resources. We recommend
 that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. L'uso di `iam:CreateServiceLinkedRole` in una policy con l'elemento `NotResource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: crea una reflex con una stella in azione e NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in action and NotResource: Using an action with a wildcard(*) and
 NotResource can allow creation of unintended service-linked roles because it can allow
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an action with a wildcard(*) and NotResource can
 allow creation of unintended service-linked roles because it can allow
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. Le policy con un carattere jolly (*) nella `Action` che includono l'elemento `NotResource` possono consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: crea SLR con e NotAction NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with NotAction and NotResource: Using NotAction with NotResource
 can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on multiple resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using NotAction with NotResource can allow creation of unintended
 service-linked roles because it allows iam:CreateServiceLinkedRole permissions on
 multiple resources. We recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. L'uso dell'elemento `NotAction` con l'elemento `NotResource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di riscrivere la policy per consentire `iam:CreateServiceLinkedRole` su un elemento limitato di ARN nell'elemento `Resource`. È inoltre possibile aggiungere `iam:CreateServiceLinkedRole` all'elemento `NotAction`.

- CreateServiceLinkedRole operazione

- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso generale: creazione di SLR con stella nella risorsa

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in resource: Using the iam:CreateServiceLinkedRole action with
 wildcards (*) in the resource can allow creation of unintended service-linked roles.
 We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:CreateServiceLinkedRole action with wildcards (*) in
 the resource can allow creation of unintended service-linked roles. We recommend that
 you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. L'uso di `iam:CreateServiceLinkedRole` in una policy con un carattere jolly (*) l'elemento `Resource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`.

- CreateServiceLinkedRole operazione
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso generale

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso riguardano utenti esperti all'interno del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso ai power user e concedono le autorizzazioni per creare ruoli

collegati ai servizi per qualsiasi servizio. AWS AWS consiglia di collegare le seguenti policy AWS gestite solo alle identità IAM che consideri power user.

- PowerUserAccess
- AlexaForBusinessFullAccess
- AWSOrganizationsServiceTrustPolicy— Questa policy AWS gestita fornisce le autorizzazioni per l'utilizzo da parte del ruolo collegato al AWS Organizations servizio. Questo ruolo consente alle Organizzazioni di creare ruoli aggiuntivi collegati ai servizi per altri servizi dell'organizzazione AWS .

Avviso generale: creazione di SLR con stella nell'operazione e nella risorsa

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in action and resource: Using wildcards (*) in the action and
 the resource can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using wildcards (*) in the action and the resource
 can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. Le policy con un carattere jolly (*) negli elementi `Action` e `Resource` possono consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. Ciò consente di creare un ruolo collegato al servizio quando si specifica`"Action": "*"`, `"Action": "iam:*"` o. `"Action": "iam:Create*"` AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento.

- CreateServiceLinkedRole operazione
- Elementi delle policy JSON IAM: Action

- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso generale

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso sono destinati agli amministratori del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso all'amministratore e concedono le autorizzazioni per creare ruoli collegati ai servizi per qualsiasi servizio. AWS AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM che consideri amministratori.

- AdministratorAccess
- IAM FullAccess

Avviso generale: crea una reflex con una stella nella risorsa e NotAction

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Create SLR with star in resource and NotAction: Using a resource with wildcards (*)
 and NotAction can allow creation of unintended service-linked roles because it allows
 iam:CreateServiceLinkedRole permissions on all resources. We recommend that you
 specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a resource with wildcards (*) and NotAction can allow creation
 of unintended service-linked roles because it allows iam:CreateServiceLinkedRole
 permissions on all resources. We recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso generale

L'azione `iam:CreateServiceLinkedRole` concede l'autorizzazione a creare un ruolo IAM che consente a un AWS servizio di eseguire azioni per tuo conto. L'uso dell'elemento `NotAction` in una policy con un carattere jolly (*) nell'elemento `Resource` può consentire la creazione di ruoli collegati ai servizi non intenzionali per più risorse. AWS consiglia invece di specificare gli ARN consentiti nell'elemento `Resource`. È inoltre possibile aggiungere `iam:CreateServiceLinkedRole` all'elemento `NotAction`.

- CreateServiceLinkedRole operazione
- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

Avviso generale: chiave di condizione globale obsoleta

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Deprecated global condition key: We recommend that you update aws:ARN to use the newer
 condition key aws:PrincipalArn.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you update aws:ARN to use the newer condition key
 aws:PrincipalArn."
```

Risoluzione dell'avviso generale

La policy include una chiave di condizione globale obsoleta. Aggiorna la chiave di condizione nella coppia chiave-valore della condizione per utilizzare una chiave di condizione globale supportata.

- Chiavi della condizione globale

Avviso generale: valore data non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid date value: The date {{date}} might not resolve as expected. We recommend that
 you use the YYYY-MM-DD format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The date {{date}} might not resolve as expected. We recommend that
 you use the YYYY-MM-DD format."
```

Risoluzione dell'avviso generale

Il tempo Unix Epoch descrive un punto nel tempo trascorso dal 1 gennaio 1970, meno i secondi intercalari. L'ora dell'epoca potrebbe non corrispondere all'ora esatta prevista. AWS consiglia di utilizzare lo standard W3C per i formati di data e ora. Ad esempio, è possibile specificare una data completa, ad esempio AAAA-MM-GG (1997-07-16), oppure aggiungere l'ora al secondo, ad esempio AAAA-MM-GGTHH:MM:SSTZD (1997-07-16T 19:20:30 + 01:00).

- Formati di data e ora W3C
- Elementi delle policy JSON IAM: Version
- aws: chiave di CurrentTime condizione globale

Avviso generale: riferimento al ruolo non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid role reference: The Principal element includes the IAM role ID {{roleid}}. We
 recommend that you use a role ARN instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element includes the IAM role ID {{roleid}}. We
 recommend that you use a role ARN instead."
```

Risoluzione dell'avviso generale

AWS consiglia di specificare l'Amazon Resource Name (ARN) per un ruolo IAM anziché il relativo ID principale. Quando IAM salva la policy, trasformerà l'ARN nell'ID principale per il ruolo esistente. AWS include una precauzione di sicurezza. Se qualcuno elimina e crea nuovamente il ruolo, avrà un nuovo ID e la policy non corrisponderà all'ID del nuovo ruolo.

- Specifica di un principale: ruoli IAM
- ARN IAM
- ID univoci IAM

Avviso generale: riferimento utente non valido

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Invalid user reference: The Principal element includes the IAM user ID {{userid}}. We
 recommend that you use a user ARN instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The Principal element includes the IAM user ID {{userid}}. We
 recommend that you use a user ARN instead."
```

Risoluzione dell'avviso generale

AWS consiglia di specificare l'Amazon Resource Name (ARN) per un utente IAM anziché il suo ID principale. Quando IAM salva la policy, trasformerà l'ARN nell'ID principale per l'utente esistente. AWS include una precauzione di sicurezza. Se qualcuno elimina e crea nuovamente l'utente, avrà un nuovo ID e la policy non corrisponderà all'ID del nuovo utente.

- Specifica di un principale: utenti IAM
- ARN IAM
- ID univoci IAM

Avviso generale: versione mancante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing version: We recommend that you specify the Version element to help you with
 debugging permission issues.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you specify the Version element to help you with
 debugging permission issues."
```

Risoluzione dell'avviso generale

AWS consiglia di includere il `Version` parametro opzionale nella politica. Se non includi un elemento Versione, per impostazione predefinita il valore viene impostato su `2012-10-17`, ma le funzionalità

più recenti, come le variabili di policy, non funzioneranno con la policy. Ad esempio, le variabili tipo `${aws:username}` non saranno riconosciute come variabili e verranno trattate come stringhe letterali nella policy.

- Elementi delle policy JSON IAM: Version

Avviso generale: sid univoci consigliati

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Unique Sids recommended: We recommend that you use statement IDs that are unique to
 your policy. Update the Sid value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "We recommend that you use statement IDs that are unique to your
 policy. Update the Sid value."
```

Risoluzione dell'avviso generale

AWS consiglia di utilizzare ID di dichiarazione univoci. L'elemento `Sid` (ID istruzione) consente di immettere un identificatore opzionale fornito per l'istruzione della policy. Puoi assegnare un valore ID di istruzione a ogni istruzione in una matrice di istruzioni utilizzando l'elemento `SID`.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Avviso generale: carattere jolly senza operatore like

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Wildcard without like operator: Your condition value includes a * or ? character. If
 you meant to use a wildcard (*, ?), update the condition operator to include Like.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Your condition value includes a * or ? character. If you meant to
 use a wildcard (*, ?), update the condition operator to include Like."
```

Risoluzione dell'avviso generale

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. Quando specifichi un valore di condizione che utilizza un carattere jolly (*,?) , devi utilizzare la versione `Like`dell'operatore di condizione. Ad esempio, anziché l'operatore di condizione stringa `StringEquals`, utilizza `StringLike`.

```
"Condition": {"StringLike": {"aws:PrincipalTag/job-category": "admin-*"}}
```

- Elementi della policy JSON IAM: operatori di condizione
- Elementi della policy JSON IAM: Condition

AWS politiche gestite con questo avviso generale

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite includono i caratteri jolly nel loro valore di condizione senza un operatore di condizione che `Like` includa il pattern matching. Quando si utilizza la policy AWS gestita come riferimento per creare una policy gestita dai clienti, si AWS consiglia di utilizzare un operatore di condizione che supporti il pattern-matching con caratteri jolly (*,?) , ad esempio. `StringLike`

- AWSGlueConsoleSageMakerNotebookFullAccess

Avviso generale: la dimensione della policy supera la quota delle policy di identità

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds identity policy quota: The {{policySize}} characters in the
 identity policy, excluding whitespace, exceed the {{policySizeQuota}} character
 maximum for inline and managed policies. We recommend that you use multiple granular
 policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the identity policy, excluding
 whitespace, exceed the {{policySizeQuota}} character maximum for inline and managed
 policies. We recommend that you use multiple granular policies."
```

Risoluzione dell'avviso generale

Puoi collegare fino a 10 policy gestite a un'identità IAM (utente, gruppo di utenti o ruolo). Tuttavia, le dimensioni di ciascuna policy gestita non possono superare la quota di default i 6.144 caratteri. IAM non calcola gli spazi vuoti per determinare le dimensioni di una policy rispetto a tali limiti. Le quote, chiamate anche limiti in AWS, sono i valori massimi per le risorse, le azioni e gli elementi presenti nell'account AWS .

Inoltre, puoi aggiungere a un'identità IAM tutte le policy in linea desiderate. Tuttavia, la dimensione della somma di tutte le policy in linea per identità non può superare la quota specificata.

Se la policy è maggiore della quota, è possibile organizzare la policy in più istruzioni e raggruppare le istruzioni in più policy.

Termini correlati

- IAM e quote di AWS STS caratteri
- Istruzioni e policy multiple
- Policy gestite dal cliente IAM
- Panoramica delle policy JSON
- Sintassi della policy JSON IAM

AWS politiche gestite con questo avviso generale

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Le seguenti politiche AWS gestite concedono le autorizzazioni alle azioni su molti AWS servizi e superano la dimensione massima delle policy. Quando si utilizza la policy gestita da AWS come riferimento per creare la policy gestita, è necessario suddividerla in più policy.

- ReadOnlyAccess
- AWSSupportServiceRolePolicy

Avviso generale: la dimensione della policy supera la quota delle policy delle risorse

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Policy size exceeds resource policy quota: The {{policySize}} characters in the
 resource policy exceed the {{policySizeQuota}} character maximum for resource
 policies. We recommend that you use multiple granular policies.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{policySize}} characters in the resource policy exceed the
 {{policySizeQuota}} character maximum for resource policies. We recommend that you use
 multiple granular policies."
```

Risoluzione dell'avviso generale

Le policy basate sulle risorse sono documenti di policy JSON che colleghi a una risorsa, come ad esempio un bucket Amazon S3. Queste policy concedono all'entità principale specificata l'autorizzazione per eseguire operazioni specifiche sulla risorsa e definiscono le condizioni in cui ciò si applica. La dimensione delle policy basate sulle risorse non può superare la quota impostata per quella risorsa. Le quote, chiamate anche limiti in AWS, sono i valori massimi per le risorse, le azioni e gli elementi presenti nell'account AWS .

Se la policy è maggiore della quota, è possibile organizzare la policy in più istruzioni e raggruppare le istruzioni in più policy.

Termini correlati

- Policy basate su risorse
- policy del bucket di Amazon S3
- Istruzioni e policy multiple
- Panoramica delle policy JSON
- Sintassi della policy JSON IAM

Avviso generale: mancata corrispondenza del tipo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch: Use the operator type {{allowed}} instead of operator {{operator}} for
 the condition key {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Use the operator type {{allowed}} instead of operator {{operator}}
 for the condition key {{key}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione supportato.

Ad esempio, la chiave di condizione globale di `aws:MultiFactorAuthPresent` richiede un operatore di condizione con il tipo di dati `Boolean`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: booleano con mancata corrispondenza del tipo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch Boolean: Add a valid Boolean value (true or false) for the condition
 operator {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid Boolean value (true or false) for the condition operator
 {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare un tipo di dati dell'operatore condizione booleano, ad esempio `true` o `false`.

Ad esempio, la chiave di condizione globale di `aws:MultiFactorAuthPresent` richiede un operatore di condizione con il tipo di dati `Boolean`. Se specifichi una data o un numero intero, il tipo di dati non corrisponderà.

Termini correlati

- Operatori di condizione booleani
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: data della mancata corrispondenza del tipo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch date: The date condition operator is used with an invalid value. Specify
 a valid date using YYYY-MM-DD or other ISO 8601 date/time format.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The date condition operator is used with an invalid value. Specify a
 valid date using YYYY-MM-DD or other ISO 8601 date/time format."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione data, in un formato data ora `YYYY-MM-DD` o altro ISO 8601.

Termini correlati

- Operatori di condizione data
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: numero della mancata corrispondenza del tipo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch number: Add a valid numeric value for the condition operator
 {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid numeric value for the condition operator {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione numerico.

Termini correlati

- Operatori di condizione numerici
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: stringa della mancata corrispondenza del tipo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Type mismatch string: Add a valid base64-encoded string value for the condition
 operator {{operator}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a valid base64-encoded string value for the condition operator
 {{operator}}."
```

Risoluzione dell'avviso generale

Aggiorna il testo per utilizzare il tipo di dati dell'operatore di condizione stringa.

Termini correlati

- Operatori di condizione stringa
- Elementi della policy JSON IAM: operatori di condizione

Avviso generale: si consigliano repository e ramo github specifici

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Specific github repo and branch recommended: Using a wildcard (*) in
 token.actions.githubusercontent.com:sub can allow requests from more sources than
 you intended. Specify the value of token.actions.githubusercontent.com:sub with the
 repository and branch name.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a wildcard (*) in token.actions.githubusercontent.com:sub
 can allow requests from more sources than you intended. Specify the value of
 token.actions.githubusercontent.com:sub with the repository and branch name."
```

Risoluzione dell'avviso generale

Se lo utilizzi GitHub come IdP OIDC, la best practice consiste nel limitare le entità che possono assumere il ruolo associato all'IDP IAM. Quando includi una `Condition` dichiarazione in una policy sulla fiducia dei ruoli, puoi limitare il ruolo a un' GitHub organizzazione, un repository o una filiale specifici. Puoi utilizzare la chiave della condizione `token.actions.githubusercontent.com:sub` per limitare l'accesso. Ti consigliamo di limitare la condizione a un insieme specifico di repository o rami. Se utilizzi un wildcard (*) in`token.actions.githubusercontent.com:sub`, GitHub le azioni provenienti da organizzazioni o repository al di fuori del tuo controllo possono assumere ruoli associati all' GitHub IdP IAM nel tuo account. AWS

Termini correlati

- Configurazione di un ruolo per il provider di identità OIDC GitHub

Avviso generale: la dimensione della policy supera la quota delle policy di attendibilità del ruolo

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Policy size exceeds role trust policy quota: The characters in the role trust policy,
 excluding whitespace, exceed the character maximum. We recommend that you request a
 role trust policy length quota increase using Service Quotas and AWS Support Center.
 If the quotas have already been increased, then you can ignore this warning.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The characters in the role trust policy, excluding whitespace,
 exceed the character maximum. We recommend that you request a role trust policy length
 quota increase using Service Quotas and AWS Support Center. If the quotas have already
 been increased, then you can ignore this warning."
```

Risoluzione dell'avviso generale

IAM e AWS STS disponiamo di quote che limitano la dimensione delle politiche di fiducia dei ruoli. I caratteri nella policy di attendibilità del ruolo, esclusi gli spazi bianchi, superano il numero massimo di caratteri. È consigliabile richiedere un aumento della quota della policy di attendibilità del ruolo utilizzando Service Quotas o AWS Support Center Console.

Termini correlati

- IAM e AWS STS quote, requisiti relativi ai nomi e limiti di caratteri

Avviso di sicurezza: consenti con NotPrincipal

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Allow with NotPrincipal: Using Allow with NotPrincipal can be overly permissive. We
 recommend that you use Principal instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using Allow with NotPrincipal can be overly permissive. We recommend
 that you use Principal instead."
```

Risoluzione dell'avviso di sicurezza

L'uso di `"Effect": "Allow"` con `NotPrincipal` può essere eccessivamente permissivo. Ad esempio, questo può concedere autorizzazioni a responsabili anonimi. AWS consiglia di specificare i principali a cui è necessario accedere utilizzando l'elemento. `Principal` In alternativa, è possibile consentire un accesso ampio e quindi aggiungere un'altra istruzione che utilizza l'elemento `NotPrincipal` con `“Effect”: “Deny”`.

- AWS Elementi delle policy JSON: principale
- AWS Elementi della policy JSON: NotPrincipal

Avviso di sicurezza: ForAllValues con chiave a valore singolo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
ForAllValues with single valued key: Using ForAllValues qualifier with the single-
valued condition key {{key}} can be overly permissive. We recommend that you remove
 ForAllValues:.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using ForAllValues qualifier with the single-valued condition key
 {{key}} can be overly permissive. We recommend that you remove ForAllValues:."
```

Risoluzione dell'avviso di sicurezza

AWS consiglia di utilizzarlo `ForAllValues` solo con condizioni multivalore. L'operatore impostato `ForAllValues` verifica se il valore di ogni membro del set di richieste è un sottoinsieme del set di chiavi di condizione. La condizione restituisce true se ogni valore delle chiavi nella richiesta corrisponde ad almeno un valore nella policy. Restituisce true anche se non ci sono chiavi nella richiesta o se i valori delle chiavi si riducono a un set di dati nullo, ad esempio una stringa vuota.

Per sapere se una condizione supporta un valore singolo o più valori, rivedi la pagina Operazioni, risorse e chiavi di condizione per il servizio. Le chiavi di condizione con il prefisso del tipo di dati `ArrayOf` sono chiavi di condizione multivalore. Ad esempio, Amazon SES supporta chiavi con valori singoli (`String`) e il tipo di dati multivalore `ArrayOfString`.

- Chiavi di contesto multivalore

Avviso di sicurezza: passa il ruolo con NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with NotResource: Using the iam:PassRole action with NotResource can be
 overly permissive because it can allow iam:PassRole permissions on multiple resources.
 We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:PassRole action with NotResource can be overly
 permissive because it can allow iam:PassRole permissions on multiple resources. We
 recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'utilizzo `iam:PassRole` di una policy con l'`NotResource`elemento può consentire ai responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- scopo: PassedToService
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: passa il ruolo con star in azione e NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in action and NotResource: Using an action with a wildcard (*) and
 NotResource can be overly permissive because it can allow iam:PassRole permissions on
 multiple resources. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an action with a wildcard (*) and NotResource can be overly
 permissive because it can allow iam:PassRole permissions on multiple resources. We
 recommend that you specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le policy con un carattere jolly (*) `Action` e che includono l'`NotResource`elemento

possono consentire ai tuoi responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- scopo: PassedToService
- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: passa il ruolo con e NotAction NotResource

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with NotAction and NotResource: Using NotAction with NotResource can
 be overly permissive because it can allow iam:PassRole permissions on multiple
 resources.. We recommend that you specify resource ARNs instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using NotAction with NotResource can be overly permissive because
 it can allow iam:PassRole permissions on multiple resources.. We recommend that you
 specify resource ARNs instead."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'utilizzo dell'`NotAction`elemento e l'elenco di alcune risorse nell'`NotResource`elemento possono consentire ai responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- scopo: PassedToService
- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action

- Elementi della policy IAM JSON: NotResource
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: invio del ruolo con stella in risorsa

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in resource: Using the iam:PassRole action with wildcards (*)
 in the resource can be overly permissive because it allows iam:PassRole permissions
 on multiple resources. We recommend that you specify resource ARNs or add the
 iam:PassedToService condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the iam:PassRole action with wildcards (*) in the resource can
 be overly permissive because it allows iam:PassRole permissions on multiple resources.
 We recommend that you specify resource ARNs or add the iam:PassedToService condition
 key to your statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le politiche che lo consentono `iam:PassRole` e che includono un carattere jolly (*) nell'`Resource`elemento possono consentire ai tuoi responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

Alcuni AWS servizi includono il proprio spazio dei nomi di servizio nel nome del proprio ruolo. Questo controllo delle policy tiene conto di queste convenzioni durante l'analisi della policy per generare i risultati. Ad esempio, il seguente ARN della risorsa potrebbe non generare un risultato:

```
arn:aws:iam::*:role/Service*
```

- Invio di un ruolo a un servizio
- Io sono: PassedToService
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso di sicurezza

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Uno di questi casi d'uso riguarda gli amministratori all'interno del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso all'amministratore e concedono le autorizzazioni per trasferire qualsiasi ruolo IAM a qualsiasi servizio. AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM che consideri amministratori.

- AdministratorAccess-Amplifica

Le seguenti politiche AWS gestite includono le autorizzazioni per la risorsa `iam:PassRole` con un carattere jolly (*) e si trovano in un percorso di obsolescenza. Per ciascuna di queste politiche, abbiamo aggiornato le linee guida sulle autorizzazioni, ad esempio consigliando una nuova politica AWS gestita che supporti il caso d'uso. Per visualizzare le alternative a queste policy, consulta per guide relative a ogni servizio.

- AWSElasticBeanstalkFullAccess
- AWSElasticBeanstalkService
- AWSLambdaFullAccess
- AWSLambdaReadOnlyAccess
- AWSOpsWorksFullAccess
- AWSOpsWorksRole
- AWSDataPipelineRole
- AmazonDynamoDB FullAccesswithDataPipeline
- AmazonElasticMapReduceFullAccess
- AmazonDynamoDB FullAccesswithDataPipeline
- Amazon EC2 ContainerServiceFullAccess

Le seguenti politiche AWS gestite forniscono autorizzazioni solo per i ruoli collegati ai servizi, che consentono ai AWS servizi di eseguire azioni per tuo conto. Non puoi collegare queste policy alle identità IAM.

- AWSServiceRoleForAmazonEKSNodegroup

Avviso di sicurezza: invio del ruolo con stella in azione e risorsa

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in action and resource: Using wildcards (*) in the action and the
 resource can be overly permissive because it allows iam:PassRole permissions on all
 resources. We recommend that you specify resource ARNs or add the iam:PassedToService
 condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using wildcards (*) in the action and the resource can be overly
 permissive because it allows iam:PassRole permissions on all resources. We recommend
 that you specify resource ARNs or add the iam:PassedToService condition key to your
 statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). Le policy con un carattere jolly (*) negli `Resource` elementi `Action` and possono consentire ai tuoi responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- scopo: PassedToService
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo avviso di sicurezza

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi AWS d'uso generali.

Alcuni di questi casi d'uso sono destinati agli amministratori del tuo account. Le seguenti politiche AWS gestite forniscono l'accesso all'amministratore e concedono le autorizzazioni per trasferire

qualsiasi ruolo IAM a qualsiasi servizio. AWS AWS consiglia di allegare le seguenti politiche AWS gestite solo alle identità IAM che consideri amministratori.

- AdministratorAccess
- IAM FullAccess

Avviso di sicurezza: assegna il ruolo di star nelle risorse e NotAction

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Pass role with star in resource and NotAction: Using a resource with wildcards (*) and
 NotAction can be overly permissive because it allows iam:PassRole permissions on all
 resources. We recommend that you specify resource ARNs or add the iam:PassedToService
 condition key to your statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using a resource with wildcards (*) and NotAction can be overly
 permissive because it allows iam:PassRole permissions on all resources. We recommend
 that you specify resource ARNs or add the iam:PassedToService condition key to your
 statement."
```

Risoluzione dell'avviso di sicurezza

Per configurare molti AWS servizi, è necessario passare un ruolo IAM al servizio. Per consentire questo è necessario concedere l'autorizzazione `iam:PassRole` a un'identità (utente, gruppo di utenti o ruolo). L'utilizzo dell'`NotAction`elemento in una policy con un carattere jolly (*) nell'`Resource`elemento può consentire ai responsabili di accedere a più servizi o funzionalità di quanto previsto. AWS consiglia invece di specificare gli ARN consentiti nell'`Resource`elemento. Inoltre, è possibile ridurre le autorizzazioni per un singolo servizio utilizzando la chiave di condizione `iam:PassedToService`.

- Invio di un ruolo a un servizio
- scopo: PassedToService
- Elementi della policy IAM JSON: NotAction
- Elementi delle policy JSON IAM: Action
- Elementi delle policy JSON IAM: Resource

Avviso di sicurezza: chiavi di condizione abbinate mancanti

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing paired condition keys: Using the condition key {{conditionKeyName}}
 can be overly permissive without also using the following condition keys:
 {{recommendedKeys}}. Condition keys like this one are more secure when paired with
 a related key. We recommend that you add the related condition keys to the same
 condition block.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the condition key {{conditionKeyName}} can be overly
 permissive without also using the following condition keys: {{recommendedKeys}}.
 Condition keys like this one are more secure when paired with a related key. We
 recommend that you add the related condition keys to the same condition block."
```

Risoluzione dell'avviso di sicurezza

Alcune chiavi di condizione sono più sicure se abbinate ad altre chiavi di condizione correlate. AWS consiglia di includere le chiavi di condizione correlate nello stesso blocco di condizione della chiave di condizione esistente. Ciò rende più sicure le autorizzazioni concesse tramite la policy.

Ad esempio, è possibile utilizzare la chiave di condizione `aws:VpcSourceIp` per confrontare l'indirizzo IP da cui è stata effettuata una richiesta con l'indirizzo IP specificato nella policy. AWS consiglia di aggiungere la chiave di condizione `aws:SourceVPC` correlata. Controlla se la richiesta proviene dal VPC specificato nella policy e l'indirizzo IP specificato.

Termini correlati

- Chiave della condizione globale di `aws:VpcSourceIp`
- Chiave della condizione globale di `aws:SourceVPC`
- Chiavi della condizione globale
- Elemento condizione
- Panoramica delle policy JSON

Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Deny with unsupported tag condition key for service: Using the effect Deny with the
 tag condition key {{conditionKeyName}} and actions for services with the following
 prefixes can be overly permissive: {{serviceNames}}. Actions for the listed services
 are not denied by this statement. We recommend that you move these actions to a
 different statement without this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Deny with the tag condition key
 {{conditionKeyName}} and actions for services with the following prefixes can be
 overly permissive: {{serviceNames}}. Actions for the listed services are not denied
 by this statement. We recommend that you move these actions to a different statement
 without this condition key."
```

Risoluzione dell'avviso di sicurezza

L'utilizzo di chiavi di condizione dei tag non supportate nell'`Condition`elemento di una policy with `"Effect": "Deny"` può essere eccessivamente permissivo, poiché la condizione viene ignorata per quel servizio. AWS consiglia di rimuovere le azioni di servizio che non supportano la chiave di condizione e di creare un'altra istruzione per negare l'accesso a risorse specifiche per tali azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` e non è supportata da un'operazione di servizio, la chiave non viene inclusa nel contesto della richiesta. In questo caso, la condizione nell'istruzione `Deny` restituisce sempre `false` e l'operazione non viene mai negata. Ciò accade anche se la risorsa è taggata correttamente.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

> **Note**
>
> Alcuni servizi consentono il supporto per la chiave di condizione `aws:ResourceTag` per un sottoinsieme di risorse e operazioni. Sistema di analisi degli accessi AWS IAM restituisce risultati per le operazioni di servizio non supportate. Ad esempio, Amazon S3 supporta `aws:ResourceTag` per un sottoinsieme delle relative risorse. Per visualizzare tutti i tipi di

risorse disponibili in Amazon S3 che supportano la chiave di condizione `aws:ResourceTag`, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Ad esempio, supponiamo che tu desideri rifiutare l'accesso per rimuovere tag da risorse specifiche che sono taggate con la coppia chiave-valore `status=Confidential`. Supponiamo inoltre che ciò AWS Lambda consenta di etichettare e rimuovere i tag dalle risorse, ma non supporti la chiave di `aws:ResourceTag` condizione. Per negare le azioni di eliminazione per AWS App Mesh e AWS Backup se questo tag è presente, usa il tasto `aws:ResourceTag` condition. Per Lambda, utilizza una convenzione di denominazione delle risorse che include il prefisso `"Confidential"`. Quindi includi un'istruzione separata che impedisca l'eliminazione delle risorse con tale convenzione di denominazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyDeleteSupported",
            "Effect": "Deny",
            "Action": [
                "appmesh:DeleteMesh",
                "backup:DeleteBackupPlan"
                ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/status": "Confidential"
                }
            }
        },
        {
            "Sid": "DenyDeleteUnsupported",
            "Effect": "Deny",
            "Action": "lambda:DeleteFunction",
            "Resource": "arn:aws:lambda:*:123456789012:function:status-Confidential*"
        }
    ]
}
```

> **Warning**
>
> Non utilizzare la IfExistsversione... dell'operatore di condizione come soluzione alternativa per questo risultato. Questo significa "Rifiuta l'operazione se la chiave è presente nel contesto della richiesta e i valori corrispondono. Altrimenti, rifiuta l'operazione". Nell'esempio precedente, inclusa l'operazione `lambda:DeleteFunction` nell'istruzione `DenyDeleteSupported` con l'operatore `StringEqualsIfExists` rifiuta sempre sempre l'operazione. Per tale operazione, la chiave non è presente nel contesto e ogni tentativo di eliminare tale tipo di risorsa viene negato, indipendentemente dal fatto che la risorsa sia taggata o meno.

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Avviso di sicurezza: nega NotAction con tag non supportato (chiave di condizione per il servizio)

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Deny NotAction with unsupported tag condition key for service: Using the effect Deny
 with NotAction and the tag condition key {{conditionKeyName}} can be overly permissive
 because some service actions are not denied by this statement. This is because the
 condition key doesn't apply to some service actions. We recommend that you use Action
 instead of NotAction.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Deny with NotAction and the tag condition key
 {{conditionKeyName}} can be overly permissive because some service actions are not
 denied by this statement. This is because the condition key doesn't apply to some
 service actions. We recommend that you use Action instead of NotAction."
```

Risoluzione dell'avviso di sicurezza

L'uso delle chiavi di condizione dei tag nell'elemento `Condition` di una policy con l'elemento `NotAction` e `"Effect": "Deny"` può essere eccessivamente permissivo. La condizione viene ignorata per le azioni di servizio che non supportano la chiave di condizione. AWS consiglia di riscrivere la logica per negare un elenco di azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` con `NotAction`, tutte le operazioni di servizio nuove o esistenti che non supportano la chiave non vengono rifiutate. AWS consiglia di elencare esplicitamente le operazioni che si desidera negare. Sistema di analisi degli accessi AWS IAM restituisce una ricerca separata per le operazioni elencate che non supportano la chiave di condizione `aws:ResourceTag`. Per ulteriori informazioni, consulta Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Avviso di sicurezza: limita l'accesso al principale del servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Restrict access to service principal: Granting access to a service principal without
 specifying a source is overly permissive. Use aws:SourceArn, aws:SourceAccount,
 aws:SourceOrgID, or aws:SourceOrgPaths condition key to grant fine-grained access.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Granting access to a service principal without specifying a source
 is overly permissive. Use aws:SourceArn, aws:SourceAccount, aws:SourceOrgID, or
 aws:SourceOrgPaths condition key to grant fine-grained access."
```

Risoluzione dell'avviso di sicurezza

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un service principal, che è un identificatore del servizio. Quando concedi l'accesso a un principale del servizio per agire per conto tuo, limita l'accesso. È possibile evitare politiche eccessivamente permissive utilizzando le chiavi`aws:SourceArn`, `aws:SourceAccount`aws:SourceOrgID, o `aws:SourceOrgPaths` condition per limitare l'accesso a una fonte specifica, ad esempio l'ARN di una risorsa specifica, l'ID dell'organizzazione o i percorsi Account AWS dell'organizzazione. La limitazione dell'accesso consente di prevenire un problema di sicurezza chiamato problema del "confused deputy".

Termini correlati

- Servizio AWS presidi
- AWS chiavi di condizione globali: aws: SourceAccount
- AWS chiavi di condizione globali: aws: SourceArn
- AWS chiavi di condizione globali: aws: SourceOrgId
- AWS chiavi di condizione globali: aws: SourceOrgPaths
- Problema del "confused deputy"

Avviso di sicurezza: chiavi di condizione mancante per il principale oidc

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing condition key for oidc principal: Using an Open ID Connect principal without a
 condition can be overly permissive. Add condition keys with a prefix that matches your
 federated OIDC principals to ensure that only the intended identity provider assumes
 the role.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using an Open ID Connect principal without a condition can be
 overly permissive. Add condition keys with a prefix that matches your federated OIDC
 principals to ensure that only the intended identity provider assumes the role."
```

Risoluzione dell'avviso di sicurezza

L'utilizzo di un principale Open ID Connect senza una condizione può essere eccessivamente permissivo. Aggiungi chiavi di condizione con un prefisso che corrisponda ai principali OIDC federati per assicurarti che solo il provider di identità previsto assuma il ruolo.

Termini correlati

- Creazione di un ruolo per la federazione di identità Web oppure OpenID Connect (console)

Avviso di sicurezza: chiavi di condizione repository github mancanti

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Missing github repo condition key: Granting a federated GitHub principal permissions
 without a condition key can allow more sources to assume the role than you intended.
 Add the token.actions.githubusercontent.com:sub condition key and specify the branch
 and repository name in the value.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Granting a federated GitHub principal permissions without a
 condition key can allow more sources to assume the role than you intended. Add the
 token.actions.githubusercontent.com:sub condition key and specify the branch and
 repository name in the value."
```

Risoluzione dell'avviso di sicurezza

Se lo utilizzi GitHub come IdP OIDC, la best practice consiste nel limitare le entità che possono assumere il ruolo associato all'IDP IAM. Quando includi una `Condition` dichiarazione in una policy sulla fiducia dei ruoli, puoi limitare il ruolo a un' GitHub organizzazione, un repository o una filiale specifici. Puoi utilizzare la chiave della condizione `token.actions.githubusercontent.com:sub` per limitare l'accesso. Ti consigliamo di limitare la condizione a un insieme specifico di repository o rami. Se non includi questa condizione, GitHub

le azioni di organizzazioni o repository al di fuori del tuo controllo possono assumere ruoli associati all'IdP GitHub IAM nel AWS tuo account.

Termini correlati

- Configurazione di un ruolo per GitHub il provider di identità OIDC

Suggerimento: operazione array vuota

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Empty array action: This statement includes no actions and does not affect the policy.
 Specify actions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no actions and does not affect the policy.
 Specify actions."
```

Risoluzione del suggerimento

Le istruzioni devono includere un elemento `Action` o `NotAction` che include un insieme di azioni. Quando l'elemento è vuoto, l'istruzione della policy non fornisce autorizzazioni. Specifica le operazioni nell'elemento `Action`.

- Elementi delle policy JSON IAM: Action

Suggerimento: condizione array vuota

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition: There are no values for the condition key {{key}} and it does
 not affect the policy. Specify conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "There are no values for the condition key {{key}} and it does not
 affect the policy. Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` facoltativa richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. Quando il valore della condizione è vuoto, la condizione restituisce `true` e l'istruzione della policy non fornisce autorizzazioni. Specifica un valore di condizione.

- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota ForAllValues

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition ForAllValues: The ForAllValues prefix with an empty condition key
 matches only if the key {{key}} is missing from the request context. To determine if
 the request context is empty, we recommend that you use the Null condition operator
 with the value of true instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The ForAllValues prefix with an empty condition key matches only
 if the key {{key}} is missing from the request context. To determine if the request
 context is empty, we recommend that you use the Null condition operator with the value
 of true instead."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. L'operatore impostato `ForAllValues` verifica se il valore di ogni membro del set di richieste è un sottoinsieme del set di chiavi di condizione.

Quando si utilizza `ForAllValues` con una chiave di condizione vuota, la condizione corrisponde solo se non ci sono chiavi nella richiesta. AWS consiglia, se si desidera verificare se un contesto di richiesta è vuoto, di utilizzare invece l'operatore di condizione `Null`.

- Chiavi di contesto multivalore
- Operatore di condizione null
- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota ForAnyValue

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition ForAnyValue: The ForAnyValue prefix with an empty condition key
 {{key}} never matches the request context and it does not affect the policy. Specify
 conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The ForAnyValue prefix with an empty condition key {{key}} never
 matches the request context and it does not affect the policy. Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore. L'operatore impostato `ForAnyValues` verifica se almeno un membro del set di valori di richiesta è corrispondente ad almeno un membro del set di valori delle chiavi di condizione.

Quando utilizzi `ForAnyValues` con una chiave di condizione vuota, la condizione non corrisponde mai. Ciò significa che la dichiarazione non ha alcun effetto sulla politica. AWS consiglia di riscrivere la condizione.

- Chiavi di contesto multivalore
- Elementi della policy JSON IAM: Condition

Suggerimento: condizione di matrice vuota IfExists

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array condition IfExists: The IfExists suffix with an empty condition key matches
 only if the key {{key}} is missing from the request context. To determine if the
 request context is empty, we recommend that you use the Null condition operator with
 the value of true instead.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The IfExists suffix with an empty condition key matches only if the
 key {{key}} is missing from the request context. To determine if the request context
 is empty, we recommend that you use the Null condition operator with the value of true
 instead."
```

Risoluzione del suggerimento

Il suffisso `...IfExists` modifica un operatore di condizione. Ciò significa che se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere elaborata come specificato nella policy. Se la chiave non è presente, l'elemento della condizione viene valutato come true (VERO).

Quando si utilizza `...IfExists` con una chiave di condizione vuota, la condizione corrisponde solo se non ci sono chiavi nella richiesta. AWS consiglia, se si desidera verificare se un contesto di richiesta è vuoto, di utilizzare invece l'operatore di condizione `Null`.

- ... IfExists operatori di condizionamento
- Elementi della policy JSON IAM: Condition

Suggerimento: principale array vuoto

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array principal: This statement includes no principals and does not affect the
 policy. Specify principals.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no principals and does not affect the
 policy. Specify principals."
```

Risoluzione del suggerimento

È necessario utilizzare l'elemento `Principal` o `NotPrincipal` nelle policy di attendibilità per i ruoli IAM e nelle policy basate sulle risorse. Le policy basate su risorse sono policy che vengono incorporate direttamente in una risorsa.

Quando si fornisce un array vuoto nell'`Principal`elemento di un'istruzione, l'istruzione non ha alcun effetto sulla politica. AWS consiglia di specificare i responsabili che devono avere accesso alla risorsa.

- Elementi delle policy JSON IAM: Principal
- Elementi della policy IAM JSON: NotPrincipal

Suggerimento: risorsa array vuota

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty array resource: This statement includes no resources and does not affect the
 policy. Specify resources.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no resources and does not affect the policy.
 Specify resources."
```

Risoluzione del suggerimento

Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`.

Quando si fornisce un array vuoto nell'elemento risorsa di un'istruzione, l'istruzione non ha alcun effetto sulla politica. AWS consiglia di specificare Amazon Resource Names (ARN) per le risorse.

- Elementi delle policy JSON IAM: Resource
- Elementi della policy IAM JSON: NotResource

Suggerimento: condizione oggetto vuota

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty object condition: This condition block is empty and it does not affect the
 policy. Specify conditions.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This condition block is empty and it does not affect the policy.
 Specify conditions."
```

Risoluzione del suggerimento

La struttura dell'elemento `Condition` richiede l'utilizzo di un operatore di condizione e di una coppia chiave-valore.

Quando si specifica un oggetto vuoto nell'elemento di condizione di un'istruzione, l'istruzione non ha alcun effetto sulla policy. Rimuovi l'elemento facoltativo o specifica le condizioni.

- Elementi della policy JSON IAM: Condition

Suggerimento: principale oggetto vuoto

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty object principal: This statement includes no principals and does not affect the
 policy. Specify principals.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "This statement includes no principals and does not affect the
 policy. Specify principals."
```

Risoluzione del suggerimento

È necessario utilizzare l'elemento `Principal` o `NotPrincipal` nelle policy di attendibilità per i ruoli IAM e nelle policy basate sulle risorse. Le policy basate su risorse sono policy che vengono incorporate direttamente in una risorsa.

Quando si fornisce un oggetto vuoto nell'`Principal`elemento di un'istruzione, l'istruzione non ha alcun effetto sulla politica. AWS consiglia di specificare i responsabili che devono avere accesso alla risorsa.

- Elementi delle policy JSON IAM: Principal
- Elementi della policy IAM JSON: NotPrincipal

Suggerimento: valore sid vuoto

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Empty Sid value: Add a value to the empty string in the Sid element.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Add a value to the empty string in the Sid element."
```

Risoluzione del suggerimento

L'elemento `Sid` (ID istruzione) facoltativo consente di immettere un identificatore fornito per l'istruzione della policy. Puoi assegnare un valore `Sid` a ogni istruzione in un array di istruzioni. Se scegli di utilizzare l'elemento `Sid`, devi fornire un valore di stringa.

Termini correlati

- Elementi delle policy JSON IAM: Sid

Suggerimento: migliora l'intervallo IP

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Improve IP range: The non-zero bits in the IP address after the masked bits are
 ignored. Replace address with {{addr}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The non-zero bits in the IP address after the masked bits are
 ignored. Replace address with {{addr}}."
```

Risoluzione del suggerimento

Le condizioni dell'indirizzo IP devono essere nel formato CIDR standard, ad esempio 203.0.113.0/24 o 2001:DB8:1234:5678::/64. Quando si includono bit diversi da zero dopo i bit mascherati, questi non vengono presi in considerazione per la condizione. AWS consiglia di utilizzare il nuovo indirizzo incluso nel messaggio.

- Operatori di condizione indirizzo IP

- Elementi della policy JSON IAM: Condition

Suggerimento: null con qualificatore

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Null with qualifier: Avoid using the Null condition operator with the ForAllValues or
 ForAnyValue qualifiers because they always return a true or false respectively.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Avoid using the Null condition operator with the ForAllValues or
 ForAnyValue qualifiers because they always return a true or false respectively."
```

Risoluzione del suggerimento

Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali ("uguale a", "minore di" e così via) per confrontare le chiavi e i valori della policy rispetto alle chiavi e ai valori del contesto della richiesta. Per le richieste che includono più valori per una singola chiave di condizione, è necessario utilizzare gli operatori su set `ForAllValues` o `ForAnyValue`.

Quando si utilizza l'operatore di condizione `Null` con `ForAllValues`, l'istruzione restituisce sempre `true`. Quando si utilizza l'operatore di `Null` condizione con`ForAnyValue`, l'istruzione restituisce `false` sempre. AWS consiglia di utilizzare l'operatore di `StringLike` condizione con questi operatori di set.

Termini correlati

- Chiavi di contesto multivalore
- Operatore di condizione null
- Elemento condizione

Suggerimento: sottoinsieme di indirizzi IP privati

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Private IP address subset: The values for condition key aws:SourceIp include a mix
 of private and public IP addresses. The private addresses will not have the desired
```

```
 effect. aws:SourceIp works only for public IP address ranges. To define permissions
 for private IP ranges, use aws:VpcSourceIp.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include a mix of private
 and public IP addresses. The private addresses will not have the desired effect.
 aws:SourceIp works only for public IP address ranges. To define permissions for
 private IP ranges, use aws:VpcSourceIp."
```

Risoluzione del suggerimento

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici.

Se il tuo elemento `Condition` include un mix di indirizzi IP privati e pubblici, l'istruzione potrebbe non avere l'effetto desiderato. Non puoi specificare indirizzi IP privati utilizzando `aws:VpcSourceIP`.

> **Note**
>
> La chiave di condizione globale `aws:VpcSourceIP` corrisponde solo se la richiesta proviene dall'indirizzo IP specificato e passa attraverso un endpoint VPC.

- aws: chiave di condizione SourceIp globale
- aws: chiave di condizione VpcSourceIp globale
- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: Condition

Suggerimento: sottoinsieme privato NotIpAddress

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Private NotIpAddress subset: The values for condition key aws:SourceIp include a mix of
 private and public IP addresses. The private addresses have no effect. aws:SourceIp
 works only for public IP address ranges. To define permissions for private IP ranges,
 use aws:VpcSourceIp.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The values for condition key aws:SourceIp include a mix of private
 and public IP addresses. The private addresses have no effect. aws:SourceIp works
 only for public IP address ranges. To define permissions for private IP ranges, use
 aws:VpcSourceIp."
```

Risoluzione del suggerimento

La chiave di condizione globale `aws:SourceIp` funziona solo per intervalli di indirizzi IP pubblici.

Se il tuo elemento `Condition` include l'operatore di condizione `NotIpAddress` e un mix di indirizzi IP privati e pubblici, l'istruzione potrebbe non avere l'effetto desiderato. Ogni indirizzo IP pubblico non specificato nella policy corrisponderà. Nessun indirizzo IP privato corrisponderà. Per ottenere questo effetto, puoi usare `NotIpAddress` con `aws:VpcSourceIP` e specificare gli indirizzi IP privati che non devono corrispondere.

- aws: chiave di condizione SourceIp globale
- aws: chiave di condizione VpcSourceIp globale
- Operatori di condizione indirizzo IP
- Elementi della policy JSON IAM: Condition

Suggerimento: azione ridondante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Redundant action: The {{redundantActionCount}} action(s) are redundant because they
 provide similar permissions. Update the policy to remove the redundant action such as:
 {{redundantAction}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{redundantActionCount}} action(s) are redundant because they
 provide similar permissions. Update the policy to remove the redundant action such as:
 {{redundantAction}}."
```

Risoluzione del suggerimento

Quando si utilizzano i caratteri jolly (*) nell'`Action`elemento, è possibile includere autorizzazioni ridondanti. AWS consiglia di rivedere la politica e di includere solo le autorizzazioni necessarie. In questo modo è possibile rimuovere le operazioni ridondanti.

Ad esempio, le operazioni riportate di seguito includono due volte l'operazione `iam:GetCredentialReport`.

```
"Action": [
        "iam:Get*",
        "iam:List*",
        "iam:GetCredentialReport"
    ],
```

In questo esempio, le autorizzazioni sono definite per ogni operazione IAM che inizia con `Get` o `List`. Quando IAM aggiunge ulteriori operazioni get o list, questa policy le consentirà. Potresti voler consentire tutte queste azioni di sola lettura. L'operazione `iam:GetCredentialReport` è già inclusa come parte di `iam:Get*`. Per rimuovere le autorizzazioni duplicate, puoi rimuovere `iam:GetCredentialReport`.

Quando tutti i contenuti di un'operazione sono ridondanti, viene visualizzato un risultato per questo controllo delle policy. In questo esempio, se l'elemento includeva `iam:*CredentialReport`, non è considerato ridondante. Ciò include `iam:GetCredentialReport`, che è ridondante, e `iam:GenerateCredentialReport`, che non lo è. La rimozione di `iam:Get*` o `iam:*CredentialReport` modificherebbe le autorizzazioni della policy.

- Elementi delle policy JSON IAM: Action

AWS politiche gestite con questo suggerimento

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi d'uso generali AWS .

Le operazioni ridondanti non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una policy AWS gestita come riferimento per creare una policy gestita dai clienti, si AWS consiglia di rimuovere le azioni ridondanti dalla policy.

Suggerimento: valore condizione ridondante num

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Redundant condition value num: Multiple values in {{operator}} are redundant. Replace
 with the {{greatest/least}} single value for {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Multiple values in {{operator}} are redundant. Replace with the
 {{greatest/least}} single value for {{key}}."
```

Risoluzione del suggerimento

Quando si utilizzano operatori di condizioni numeriche per valori simili in una chiave di condizione, è possibile creare una sovrapposizione che si traduce in autorizzazioni ridondanti.

Ad esempio, il seguente elemento `Condition` include più condizioni `aws:MultiFactorAuthAge` che hanno una sovrapposizione di età di 1200 secondi.

```
"Condition": {
        "NumericLessThan": {
          "aws:MultiFactorAuthAge": [
            "2700",
            "3600"
          ]
        }
      }
```

In questo esempio, le autorizzazioni vengono definite se l'autenticazione a più fattori (MFA) è stata completata meno di 3600 secondi (1 ora) fa. È possibile rimuovere il valore `2700` ridondante.

- Operatori di condizione numerici
- Elementi della policy JSON IAM: Condition

Suggerimento: risorsa ridondante

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Redundant resource: The {{redundantResourceCount}} resource ARN(s) are redundant
 because they reference the same resource. Review the use of wildcards (*)
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The {{redundantResourceCount}} resource ARN(s) are redundant because
 they reference the same resource. Review the use of wildcards (*)"
```

Risoluzione del suggerimento

Quando usi i caratteri jolly (*) in Amazon Resource Name (ARN), puoi creare autorizzazioni per le risorse ridondanti.

Ad esempio, il seguente elemento `Resource` include più ARN con autorizzazioni ridondanti.

```
"Resource": [
            "arn:aws:iam::111122223333:role/jane-admin",
            "arn:aws:iam::111122223333:role/jane-s3only",
            "arn:aws:iam::111122223333:role/jane*"
        ],
```

In questo esempio, le autorizzazioni sono definite per qualsiasi ruolo con un nome che inizia con `jane`. È possibile rimuovere gli ARN `jane-admin` e `jane-s3only` ridondanti senza modificare le autorizzazioni risultanti. Questo rende la policy dinamica. Definirà le autorizzazioni per tutti i ruoli futuri che iniziano con `jane`. Se l'intenzione della policy è consentire l'accesso a un numero statico di ruoli, rimuovere l'ultimo ARN ed elencare solo gli ARN da definire.

- Elementi delle policy JSON IAM: Resource

AWS politiche gestite con questo suggerimento

AWS le politiche gestite consentono di iniziare con l'assegnazione AWS di autorizzazioni in base a casi d'uso generali AWS .

Le operazioni ridondanti non influenzano le autorizzazioni concesse dalla policy. Quando si utilizza una policy AWS gestita come riferimento per creare una policy gestita dai clienti, si AWS consiglia di rimuovere le risorse ridondanti dalla policy.

Suggerimento: istruzione ridondante

Nel AWS Management Console, i risultati di questo controllo includono il seguente messaggio:

```
Redundant statement: The statements are redundant because they provide identical
 permissions. Update the policy to remove the redundant statement.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The statements are redundant because they provide identical
 permissions. Update the policy to remove the redundant statement."
```

Risoluzione del suggerimento

L'elemento `Statement` è l'elemento principale per una policy. Questa elemento è obbligatorio. L'elemento `Statement` può contenere una singola istruzione o una matrice di singole istruzioni.

Quando si include la stessa istruzione più di una volta in una policy lunga, le istruzioni sono ridondanti. È possibile rimuovere una delle istruzioni senza influire sulle autorizzazioni concesse dalla policy. Quando un utente modifica una policy, potrebbe modificare una delle istruzioni senza aggiornare il duplicato. Ciò potrebbe comportare un numero di autorizzazioni maggiore del previsto.

- Elementi delle policy IAM JSON: istruzione

Suggerimento: carattere jolly nel nome del servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Wildcard in service name: Avoid using wildcards (*, ?) in the service name because it
 might grant unintended access to other AWS services with similar names.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Avoid using wildcards (*, ?) in the service name because it might
 grant unintended access to other AWS services with similar names."
```

Risoluzione del suggerimento

Quando si include il nome di un AWS servizio in una policy, si AWS consiglia di non includere caratteri jolly (*,?). Ciò potrebbe aggiungere autorizzazioni per servizi futuri non previsti. Ad esempio, esistono più di una dozzina di AWS servizi il cui nome contiene la parola`*code*`.

```
"Resource": "arn:aws:*code*::111122223333:*"
```

- Elementi delle policy JSON IAM: Resource

Suggerimento: consenti con chiave di condizione tag non supportata per il servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Allow with unsupported tag condition key for service: Using the effect Allow with the
 tag condition key {{conditionKeyName}} and actions for services with the following
 prefixes does not affect the policy: {{serviceNames}}. Actions for the listed service
 are not allowed by this statement. We recommend that you move these actions to a
 different statement without this condition key.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Allow with the tag condition key
 {{conditionKeyName}} and actions for services with the following prefixes does not
 affect the policy: {{serviceNames}}. Actions for the listed service are not allowed
 by this statement. We recommend that you move these actions to a different statement
 without this condition key."
```

Risoluzione del suggerimento

L'utilizzo di chiavi di condizione dei tag non supportate nell'`Condition`elemento di una policy con non `"Effect": "Allow"` influisce sulle autorizzazioni concesse dalla policy, poiché la condizione viene ignorata per quell'azione di servizio. AWS consiglia di rimuovere le azioni per i servizi che non supportano la chiave di condizione e di creare un'altra istruzione per consentire l'accesso a risorse specifiche di quel servizio.

Se utilizzi la chiave di condizione `aws:ResourceTag` e non è supportata da un'operazione di servizio, la chiave non viene inclusa nel contesto della richiesta. In questo caso, la condizione nell'istruzione `Allow` restituisce sempre `false` e l'operazione non viene mai rifiutata. Ciò accade anche se la risorsa è taggata correttamente.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato

su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

> **Note**
>
> Alcuni servizi consentono il supporto per la chiave di condizione `aws:ResourceTag` per un sottoinsieme di risorse e operazioni. Sistema di analisi degli accessi AWS IAM restituisce risultati per le operazioni di servizio non supportate. Ad esempio, Amazon S3 supporta `aws:ResourceTag` per un sottoinsieme delle relative risorse. Per visualizzare tutti i tipi di risorse disponibili in Amazon S3 che supportano la chiave di condizione `aws:ResourceTag`, consulta Tipi di risorse definiti da Amazon S3 in Service Authorization Reference.

Ad esempio, supponiamo che tu desideri consentire ai membri del team di visualizzare i dettagli per le risorse specifiche che sono taggate con la coppia chiave-valore `team=BumbleBee`. Supponiamo inoltre che ciò AWS Lambda consenta di etichettare le risorse, ma non supporti la chiave di `aws:ResourceTag` condizione. Per consentire le azioni di visualizzazione per AWS App Mesh e AWS Backup se questo tag è presente, usa il tasto `aws:ResourceTag` condition. Per Lambda, utilizza una convenzione di denominazione delle risorse che include il nome del team come prefisso. Quindi includi un'istruzione separata che consenta la visualizzazione delle risorse con tale convenzione di denominazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewSupported",
            "Effect": "Allow",
            "Action": [
                "appmesh:DescribeMesh",
                "backup:GetBackupPlan"
                ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/team": "BumbleBee"
                }
            }
        },
        {
```

```
            "Sid": "AllowViewUnsupported",
            "Effect": "Allow",
            "Action": "lambda:GetFunction",
            "Resource": "arn:aws:lambda:*:123456789012:function:team-BumbleBee*"
        }
    ]
}
```

> **Warning**
>
> Non utilizzare la `Not` versione dell'operatore di condizione con `"Effect": "Allow"` come soluzione alternativa per questo risultato. Questi operatori di condizione forniscono la corrispondenza negata. Ciò significa che dopo che la condizione è stata valutata, il risultato viene negato. Nell'esempio precedente, che include l'operazione `lambda:GetFunction` nell'istruzione `AllowViewSupported` con l'operatore `StringNotEquals` consente sempre l'operazione, indipendentemente dal fatto che la risorsa sia taggata o meno.
> Non utilizzare la IfExistsversione... dell'operatore di condizione come soluzione alternativa per questo risultato. Questo significa "Consenti l'operazione se la chiave è presente nel contesto della richiesta e i valori corrispondono. Altrimenti, autorizza l'operazione." Nell'esempio precedente, inclusa l'operazione `lambda:GetFunction` nell'istruzione `AllowViewSupported` con l'operatore `StringEqualsIfExists` consente sempre l'operazione. Per tale operazione, la chiave non è presente nel contesto e ogni tentativo di visualizzare tale tipo di risorsa viene negato, indipendentemente dal fatto che la risorsa sia taggata o meno.

Termini correlati

- Chiavi della condizione globale
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Suggerimento: consenti NotAction con tag non supportato (chiave di condizione per il servizio)

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Allow NotAction with unsupported tag condition key for service: Using the effect Allow
 with NotAction and the tag condition key {{conditionKeyName}} allows only service
 actions that support the condition key. The condition key doesn't apply to some
 service actions. We recommend that you use Action instead of NotAction.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "Using the effect Allow with NotAction and the tag condition key
 {{conditionKeyName}} allows only service actions that support the condition key. The
 condition key doesn't apply to some service actions. We recommend that you use Action
 instead of NotAction."
```

Risoluzione del suggerimento

L'uso delle chiavi di condizione dei tag non supportate nell'elemento `Condition` di una policy con l'elemento `NotAction` e `"Effect": "Allow"` non influisce sulle autorizzazioni concesse dai policy. La condizione viene ignorata per le azioni di servizio che non supportano la chiave di condizione. AWS consiglia di riscrivere la logica per consentire un elenco di azioni.

Se utilizzi la chiave di condizione `aws:ResourceTag` con `NotAction`, tutte le operazioni di servizio nuove o esistenti che non supportano la chiave non vengono rifiutate. AWS consiglia di elencare esplicitamente le operazioni che si desidera consentire. Sistema di analisi degli accessi AWS IAM restituisce una ricerca separata per le operazioni elencate che non supportano la chiave di condizione `aws:ResourceTag`. Per ulteriori informazioni, consulta Suggerimento: consenti con chiave di condizione tag non supportata per il servizio.

Quando un servizio supporta la chiave di condizione `aws:ResourceTag`, è possibile utilizzare i tag per controllare l'accesso alle risorse del servizio. Questo è noto come controllo degli accessi basato su attributi (ABAC). I servizi che non supportano queste chiavi richiedono il controllo dell'accesso alle risorse tramite il controllo degli accessi basato su risorse (RBAC).

Termini correlati

- Chiavi della condizione globale
- Confronto tra ABAC e RBAC
- Elementi della policy JSON IAM: operatori di condizione
- Elemento condizione
- Panoramica delle policy JSON

Suggerimento: chiave di condizione consigliata per il principale del servizio

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Recommended condition key for service principal: To restrict access to the service
 principal {{servicePrincipalPrefix}} operating on your behalf, we recommend
 aws:SourceArn, aws:SourceAccount, aws:SourceOrgID, or aws:SourceOrgPaths instead of
 {{key}}.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "To restrict access to the service principal
 {{servicePrincipalPrefix}} operating on your behalf, we recommend aws:SourceArn,
 aws:SourceAccount, aws:SourceOrgID, or aws:SourceOrgPaths instead of {{key}}."
```

Risoluzione del suggerimento

È possibile specificare Servizi AWS nell'`Principal`elemento di una politica basata sulle risorse utilizzando un service principal, che è un identificatore del servizio. Quando si concede l'accesso ai principali del servizio, è consigliabile utilizzare le chiavi di condizione `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` o `aws:SourceOrgPaths` anziché altre chiavi di condizione, come `aws:Referer`. Questo aiuta a prevenire un problema di sicurezza chiamato problema del "confused deputy".

Termini correlati

- Servizio AWS presidi
- AWS chiavi di condizione globali: aws: SourceAccount
- AWS chiavi di condizione globali: aws: SourceArn
- AWS chiavi di condizione globali: aws: SourceOrgId
- AWS chiavi di condizione globali: aws: SourceOrgPaths
- Problema del "confused deputy"

Suggerimento: chiave di condizione irrilevante nella policy

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Irrelevant condition key in policy: The condition key {{condition-key}} is not relevant
 for the {{resource-type}} policy.  Use this key in an identity-based policy to govern
 access to this resource.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The condition key {{condition-key}} is not relevant for the
 {{resource-type}} policy.  Use this key in an identity-based policy to govern access
 to this resource."
```

Risoluzione del suggerimento

Alcune chiavi di condizione non sono rilevanti per le policy basate sulle risorse. Ad esempio, la chiave di condizione `s3:ResourceAccount` non è rilevante per la polocy basata sulle risorse collegata a un tipo di risorsa bucket Amazon S3 o a un punto di accesso Amazon S3.

Puoi utilizzare la chiave di condizione nella policy basata sulle identità per controllare l'accesso alla risorsa.

Termini correlati

- Policy basate sulle identità e policy basate su risorse

Suggerimento: principale ridondante nella policy di attendibilità del ruolo

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Redundant principal in role trust policy: The assumed-role principal
 {{redundant_principal}} is redundant with its parent role {{parent_role}}. Remove the
 assumed-role principal.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The assumed-role principal {{redundant_principal}} is redundant with
 its parent role {{parent_role}}. Remove the assumed-role principal."
```

Risoluzione del suggerimento

Se si specifica sia un principale con ruolo assunto che il suo ruolo padre nell'elemento `Principal` di una policy, non consente o nega autorizzazioni diverse. Ad esempio, è ridondante se si specifica l'elemento `Principal` utilizzando il seguente formato:

```
"Principal": {
            "AWS": [
            "arn:aws:iam::AWS-account-ID:role/rolename",
            "arn:aws:iam::AWS-account-ID:assumed-role/rolename/rolesessionname"
        ]
```

Si consiglia di rimuovere il principale del ruolo assunto.

Termini correlati

- Principali della sessione come ruolo

Suggerimento: conferma il tipo di attestazione del pubblico

Nel AWS Management Console, il risultato di questo controllo include il seguente messaggio:

```
Confirm audience claim type: The 'aud' (audience) claim key identifies the recipients
 that the JSON web token is intended for. Audience claims can be multivalued or single-
valued. If the claim is multivalued, use a ForAllValues or ForAnyValue qualifier. If
 the claim is single-valued, do not use a qualifier.
```

Nelle chiamate programmatiche all' AWS API AWS CLI or, il risultato di questo controllo include il seguente messaggio:

```
"findingDetails": "The 'aud' (audience) claim key identifies the recipients that the
 JSON web token is intended for. Audience claims can be multivalued or single-valued.
 If the claim is multivalued, use a ForAllValues or ForAnyValue qualifier. If the claim
 is single-valued, do not use a qualifier."
```

Risoluzione del suggerimento

La chiave di attestazione `aud` (destinatario) è un identificatore univoco per l'app rilasciato durante la registrazione dell'app con IdP. Identifica i destinatari del token Web JSON. Le attestazioni del pubblico possono essere multivalore o a valore singolo. Se l'attestazione è multivalore, utilizza un'operatore di condizione `ForAllValues` o `ForAnyValue`. Se l'attestazione ha un valore singolo, non utilizzare un operatore di condizione.

Termini correlati

- Creazione di un ruolo per la federazione di identità Web oppure OpenID Connect (console)
- Chiavi di contesto multivalore
- Chiavi di condizione a valore singolo vs multivalore

Controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM

È possibile convalidare le policy rispetto agli standard di sicurezza specificati utilizzando i controlli delle policy personalizzati di AWS Identity and Access Management Access Analyzer . È possibile eseguire i seguenti tipi di controlli delle policy personalizzati:

- Verifica in base a una policy di riferimento: quando si modifica una policy, è possibile controllare se la policy aggiornata concede un nuovo accesso rispetto a quella di riferimento, ad esempio una sua versione esistente. Puoi eseguire questo controllo quando modifichi una policy utilizzando AWS Command Line Interface (AWS CLI), l'API IAM Access Analyzer (API) o l'editor di policy JSON nella console IAM.
- Verifica in base a un elenco di azioni o risorse IAM: puoi verificare che azioni o risorse IAM specifiche non siano consentite dalla tua policy. Se vengono specificate solo azioni, IAM Access Analyzer verifica l'accesso delle azioni su tutte le risorse della policy. Se vengono specificate solo risorse, IAM Access Analyzer verifica quali azioni hanno accesso alle risorse specificate. Se vengono specificate sia le azioni che le risorse, IAM Access Analyzer verifica quali delle azioni specificate hanno accesso alle risorse specificate. È possibile eseguire questo controllo quando si crea o si modifica una policy utilizzando a AWS CLI o l'API.
- Verifica l'accesso pubblico: puoi verificare se una politica delle risorse può concedere l'accesso pubblico a un tipo di risorsa specificato. È possibile eseguire questo controllo quando si crea o si modifica una politica utilizzando AWS CLI o l'API. Questo tipo di controllo delle policy personalizzate è diverso dall'anteprima dell'accesso perché non richiede alcun account o contesto di analisi degli accessi esterni. Le anteprime di accesso consentono di visualizzare in anteprima i risultati di IAM Access Analyzer prima di distribuire le autorizzazioni delle risorse, mentre il controllo personalizzato determina se l'accesso pubblico può essere concesso da una policy.

Viene addebitato un costo per ogni controllo della policy personalizzato. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

Come funzionano i controlli delle policy personalizzati

È possibile eseguire controlli delle policy personalizzati sulle policy basate su identità e risorse. I controlli delle policy personalizzati non si basano su tecniche di corrispondenza dei modelli o sulla verifica dei log di accesso per determinare se un accesso nuovo o specifico è consentito da una policy. Analogamente ai risultati degli accessi esterni, i controlli delle policy personalizzati si basano su Zelkova. Zelkova traduce le policy IAM in istruzioni logiche equivalenti e gestisce una suite di risolutori logici generici e specializzati (teorie dei moduli di soddisfacibilità) per il problema. Per verificare gli accessi nuovi o specifici, Sistema di analisi degli accessi AWS IAM applica ripetutamente Zelkova a una policy. Le query diventano sempre più specifiche per caratterizzare classi di comportamenti consentite dalla policy in base al contenuto della policy. Per ulteriori informazioni sulle teorie dei moduli di soddisfacibilità, consulta Teorie dei moduli di soddisfacibilità.

In rari casi, Sistema di analisi degli accessi AWS IAM non è in grado di determinare completamente se un'istruzione della policy concede un accesso nuovo o specifico. In questi casi, dichiara erroneamente un falso positivo non superando il controllo delle policy personalizzate. Sistema di analisi degli accessi AWS IAM è progettato per fornire una valutazione completa delle policy e si impegna per ridurre al minimo i falsi negativi. Con questo approccio, Sistema di analisi degli accessi AWS IAM garantisce in modo piuttosto certo che un controllo superato significa che l'accesso non è stato concesso dalla policy. Puoi controllare manualmente i controlli non riusciti esaminando l'istruzione della policy riportata nella risposta di Sistema di analisi degli accessi AWS IAM.

Fai riferimento agli esempi di policy per verificare la presenza di nuovi accessi

Puoi trovare esempi di policy di riferimento e scoprire come configurare ed eseguire un controllo personalizzato delle policy per nuovi accessi nell'archivio degli esempi di controlli delle policy personalizzati di IAM Access Analyzer su GitHub.

Prima di utilizzare questi esempi

Prima di utilizzare questi esempi di policy di riferimento, esegui queste operazioni:

- Esamina attentamente e personalizza le policy di riferimento per i tuoi requisiti specifici.
- Testa accuratamente le policy di riferimento nel tuo ambiente con i servizi Servizi AWS che utilizzi.

  Le policy di riferimento illustrano l'implementazione e l'utilizzo di controlli delle policy personalizzati. Non devono essere interpretate come suggerimenti o best practice AWS ufficiali da implementare esattamente come mostrato. È tua responsabilità testare

accuratamente la sostenibilità delle policy di riferimento per soddisfare i requisiti di sicurezza del tuo ambiente.

- I controlli delle policy personalizzati sono indipendenti dall'ambiente durante l'analisi. La loro analisi prende in considerazione solo le informazioni contenute nelle policy di input. Ad esempio, i controlli delle policy personalizzati non possono verificare se un account è membro di un'organizzazione specifica AWS . Pertanto, non possono confrontare i nuovi accessi in base ai valori delle chiavi di condizione per le chiavi di condizione `aws:PrincipalOrgId`e `aws:PrincipalAccount`.

Ispezione dei controlli delle policy personalizzati non riusciti

Quando un controllo delle policy personalizzate fallisce, la risposta di Sistema di analisi degli accessi AWS IAM include l'ID istruzione (`Sid`) dell'istruzione che ha causato l'esito negativo del controllo. Sebbene l'ID istruzione sia un elemento di policy facoltativo, consigliamo di aggiungere un ID istruzione per ogni istruzione di policy. Il controllo delle policy personalizzato restituisce anche un indice delle istruzioni per aiutare a identificare il motivo dell'errore del controllo. L'indice delle istruzioni segue la numerazione a base zero, in cui la prima istruzione viene indicata come 0. Quando sono presenti più istruzioni che causano l'esito negativo di un controllo, il controllo restituisce un solo ID istruzione alla volta. Consigliamo di correggere l'istruzione evidenziata nel motivo e di eseguire nuovamente il controllo finché non viene superato.

Convalida delle policy con controlli delle policy personalizzati (console)

Come passaggio facoltativo, è possibile eseguire un controllo delle policy personalizzato durante la modifica di una policy nell'editor di policy JSON nella console IAM. Puoi verificare se la policy aggiornata concede un nuovo accesso rispetto alla versione esistente.

Per verificare la presenza di nuovi accessi durante la modifica delle policy JSON IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Nell'elenco delle policy, seleziona il nome della policy che desideri modificare. Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.
4. Seleziona la scheda Autorizzazioni e scegli Modifica.
5. Scegli l'opzione JSON e aggiorna la tua policy.

6. Nel riquadro di convalida delle policy sotto la policy, scegli la scheda Verifica nuovi accessi e seleziona Verifica policy. Se le autorizzazioni modificate concedono un nuovo accesso, l'istruzione verrà evidenziata nel riquadro di convalida della policy.
7. Se non intendi concedere un nuovo accesso, aggiorna le istruzioni di policy e scegli Verifica policy finché non viene rilevato alcun nuovo accesso.

> **Note**
>
> Viene addebitato un costo per ogni controllo di un nuovo accesso. Per maggiori dettagli sui prezzi, consulta i prezzi di Sistema di analisi degli accessi AWS IAM.

8. Seleziona Successivo.
9. Nella pagina Verifica e salva, esamina il campo Autorizzazioni definite in questa policy, quindi scegli Salva modifiche.

Convalida delle politiche con controlli di policy personalizzati (AWS CLI o API)

Puoi eseguire controlli delle policy personalizzate di IAM Access Analyzer dall'API IAM Access Analyzer AWS CLI o dall'API IAM Access Analyzer.

Per eseguire controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM (AWS CLI)

- Per verificare se è consentito un nuovo accesso per una policy aggiornata rispetto alla policy esistente, esegui il seguente comando: check-no-new-access
- Per verificare se l'accesso specificato non è consentito da una policy, esegui il comando seguente: check-access-not-granted
- Per verificare se una politica delle risorse può concedere l'accesso pubblico a un tipo di risorsa specifico, esegui il comando seguente: check-no-public-access

Per eseguire controlli delle policy personalizzati di Sistema di analisi degli accessi AWS IAM (API)

- Per verificare se è consentito un nuovo accesso per una policy aggiornata rispetto alla policy esistente, utilizza l'operazione API CheckNoNewAccess.
- Per verificare se l'accesso specificato non è consentito da una policy, utilizza l'operazione API CheckAccessNotGranted.

- Per verificare se una politica delle risorse può concedere l'accesso pubblico a un tipo di risorsa specifico, utilizza l'operazione CheckNoPublicAccessAPI.

Generazione di policy per Sistema di analisi degli accessi AWS IAM

In qualità di amministratore o sviluppatore, puoi concedere autorizzazioni a entità IAM (utenti o ruoli) che vanno oltre quanto richiesto. IAM fornisce diverse opzioni che consentono di perfezionare le autorizzazioni concesse. Un'opzione consiste nel generare una policy IAM basata sull'attività di accesso per un'entità. IAM Access Analyzer esamina AWS CloudTrail i log e genera un modello di policy che contiene le autorizzazioni utilizzate dall'entità nell'intervallo di date specificato. È possibile utilizzare il modello per creare una policy con autorizzazioni granulari che concedono solo le autorizzazioni necessarie per supportare il caso d'uso specifico.

Argomenti

Come funziona la generazione di policy

IAM Access Analyzer analizza CloudTrail gli eventi per identificare le azioni e i servizi che sono stati utilizzati da un'entità IAM (utente o ruolo). Viene quindi generata una policy IAM basata su tale attività. È possibile perfezionare le autorizzazioni di un'entità quando si sostituisce una policy di autorizzazioni generali associata all'entità con la policy generata. Di seguito si riporta una panoramica di alto livello del processo di generazione della policy.

- Configurazione per la generazione di modelli di policy: specifichi un periodo di tempo fino a 90 giorni per consentire a IAM Access Analyzer di analizzare gli eventi storici. AWS CloudTrail È

necessario specificare un ruolo del servizio esistente o crearne uno nuovo. Il ruolo di servizio consente a IAM Access Analyzer di accedere al CloudTrail percorso e alle informazioni sull'ultimo accesso al servizio per identificare i servizi e le azioni utilizzati. È necessario specificare il CloudTrail percorso che registra gli eventi per l'account prima di poter generare una policy. Per ulteriori informazioni sulle quote di dati di IAM Access Analyzer, consulta le CloudTrail quote di IAM Access Analyzer.

- Genera policy: IAM Access Analyzer genera una policy basata sull'attività di accesso nei tuoi eventi. CloudTrail
- Esaminare e personalizzare la policy – Dopo la generazione della policy, è possibile esaminare i servizi e le azioni utilizzati dall'entità durante l'intervallo di date specificato. È possibile personalizzare ulteriormente la policy, aggiungendo o rimuovendo autorizzazioni, specificando risorse e aggiungendo condizioni al modello di policy.
- Creare e allegare policy – È possibile salvare la policy generata creando una policy gestita. È possibile allegare la policy creata all'utente o al ruolo la cui attività è stata utilizzata per generare la policy.

Informazioni sul servizio e sul livello di azione

Quando Sistema di analisi degli accessi AWS IAM genera una policy IAM, vengono restituite informazioni che consentono di personalizzare ulteriormente la policy. Quando viene generata una policy, è possibile restituire due categorie di informazioni:

- Policy con informazioni a livello di azione: per alcuni AWS servizi, come Amazon EC2, IAM Access Analyzer è in grado di identificare le azioni rilevate nei CloudTrail tuoi eventi ed elenca le azioni utilizzate nella policy che genera. Per un elenco dei servizi supportati, consulta Servizi di generazione di policy per Sistema di analisi degli accessi IAM. Per alcuni servizi, Sistema di analisi degli accessi AWS IAM richiede l'aggiunta azioni per i servizi alla policy generata.
- Policy con informazioni sui livelli di servizio – Sistema di analisi degli accessi AWS IAM utilizza le informazioni relative all'ultimo accesso per creare un modello di policy con tutti i servizi utilizzati di recente. Quando utilizzi AWS Management Console, ti chiediamo di esaminare i servizi e aggiungere azioni per completare la policy.

Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel riferimento di autorizzazione del servizio.

Da sapere sulla generazione di policy

Prima di generare una policy, esaminare i dettagli importanti riportati di seguito.

- Abilita un CloudTrail percorso: devi avere un CloudTrail percorso abilitato affinché il tuo account generi una politica basata sull'attività di accesso. Quando crei un CloudTrail trail, CloudTrail invia gli eventi relativi al tuo trail a un bucket Amazon S3 da te specificato. Per informazioni su come creare un CloudTrail trail, consulta Creazione di un trail per il tuo AWS account nella Guida per l'AWS CloudTrail utente.
- Eventi dati non disponibili: Sistema di analisi degli accessi AWS IAM non identifica l'attività a livello di azione per eventi di dati, ad esempio eventi di dati di Amazon S3, nelle policy generate.
- PassRole— L'`iam:PassRole`azione non viene tracciata CloudTrail e non è inclusa nelle politiche generate.
- Ridurre il tempo di generazione della policy – Per generare più rapidamente una policy, ridurre l'intervallo di date specificato durante la configurazione per la generazione delle policy.
- Utilizzo CloudTrail per il controllo: non utilizzate la generazione di policy per scopi di controllo, ma utilizzate invece. CloudTrail Per ulteriori informazioni sull'utilizzo CloudTrail, consulta Registrazione delle chiamate IAM e AWS STS API con. AWS CloudTrail
- Azioni negate: la generazione delle policy esamina tutti CloudTrail gli eventi, comprese le azioni negate.
- Una console IAM di policy – È possibile generare una policy alla volta nella console IAM.
- Console IAM per la disponibilità di policy generate – È possibile esaminare una policy generata nella console IAM per un massimo di 7 giorni dopo la sua generazione. Dopo 7 giorni, è necessario generare una nuova policy.
- Quote di generazione di policy: per ulteriori informazioni sulle quote di generazione delle policy di Sistema di analisi degli accessi AWS IAM, consulta Quote di Sistema di analisi degli accessi AWS IAM.
- Si applicano le tariffe standard di Amazon S3: quando utilizzi la funzionalità di generazione delle policy, IAM Access Analyzer esamina CloudTrail i log nel tuo bucket S3. Non sono previsti costi di archiviazione aggiuntivi per accedere ai log e generare le policy. CloudTrail AWS addebita le tariffe standard di Amazon S3 per le richieste e il trasferimento di dati dei CloudTrail log archiviati nel bucket S3.
- AWS Control Tower supporto: la generazione di policy non supporta AWS Control Tower la generazione di policy.

Autorizzazioni richieste per generare una policy

Le autorizzazioni necessarie per generare una policy per la prima volta differiscono da quelle necessarie per generare una policy per usi successivi.

Configurazioni per generare la policy la prima volta

Quando si genera una policy per la prima volta, è necessario scegliere un ruolo del servizio esistente appropriato nell'account o crearne uno nuovo. Il ruolo di servizio consente a IAM Access Analyzer di accedere alle informazioni a cui si accede per ultimo nel tuo account CloudTrail e fornisce l'ultimo accesso. Solo gli amministratori devono disporre delle autorizzazioni necessarie per creare e configurare i ruoli. Pertanto, è consigliabile che un amministratore crei il ruolo del servizio durante la prima configurazione. Per ulteriori informazioni sulle autorizzazioni necessarie per creare ruoli di servizio, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio. AWS

Autorizzazioni richieste per il ruolo del servizio

Quando si crea un ruolo del servizio, si configurano due policy per il ruolo. Si allega una policy di autorizzazioni IAM al ruolo che specifica le operazioni che il ruolo può eseguire. È inoltre possibile allegare una policy di attendibilità del ruolo al ruolo che specifica l'entità che può utilizzare il ruolo.

La prima policy di esempio mostra la policy di autorizzazioni per il ruolo del servizio necessario per generare una policy. Nella seconda policy di esempio viene illustrata la policy di attendibilità del ruolo necessaria per il ruolo del servizio. Puoi utilizzare queste politiche per aiutarti a creare un ruolo di servizio quando utilizzi l' AWS API o AWS CLI per generare una policy. Quando si utilizza la console IAM per creare un ruolo del servizio come parte del processo di generazione della policy, vengono generate automaticamente queste policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudtrail:GetTrail",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetServiceLastAccessedDetails",
```

```
                "iam:GenerateServiceLastAccessedDetails"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

La policy di esempio seguente mostra la policy di attendibilità del ruolo con le autorizzazioni che consentono a Sistema di analisi degli accessi AWS IAM di assumere il ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "access-analyzer.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Usi successivi

Per generare policy in AWS Management Console, un utente IAM deve disporre di una policy di autorizzazioni che gli consenta di trasferire il ruolo di servizio utilizzato per la generazione delle policy a IAM Access Analyzer. `iam:PassRole`di solito è accompagnata da `iam:GetRole` in modo che l'utente possa ottenere i dettagli del ruolo da assegnare. In questo esempio, l'utente può passare solo i ruoli esistenti nell'account specificato con nomi che iniziano con `AccessAnalyzerMonitorServiceRole*`. Per ulteriori informazioni sul passaggio dei ruoli IAM ai

AWS servizi, consulta Concessione a un utente delle autorizzazioni per passare un ruolo a un AWS servizio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUserToPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-role/
AccessAnalyzerMonitorServiceRole*"
    }
  ]
}
```

È inoltre necessario disporre delle seguenti autorizzazioni IAM Access Analyzer per generare policy nell' AWS API AWS Management Console, o AWS CLI come illustrato nella seguente dichiarazione di policy.

```
{
  "Sid": "AllowUserToGeneratePolicy",
  "Effect": "Allow",
  "Action": [
    "access-analyzer:CancelPolicyGeneration",
    "access-analyzer:GetGeneratedPolicy",
    "access-analyzer:ListPolicyGenerations",
    "access-analyzer:StartPolicyGeneration"
  ],
  "Resource": "*"
}
```

Per i primi utilizzi e per quelli successivi

Quando utilizzi il AWS Management Console per generare una policy, devi avere l'`cloudtrail:ListTrails`autorizzazione a elencare i CloudTrail percorsi nel tuo account, come mostrato nella seguente informativa sulla politica.

```
{
```

```
  "Sid": "AllowUserToListTrails",
  "Effect": "Allow",
  "Action": [
    "CloudTrail:ListTrails"
  ],
  "Resource": "*"
}
```

Genera una policy basata sull' CloudTrail attività (console)

È possibile generare una policy per un utente IAM o un ruolo.

Fase 1: Generare una politica basata sull' CloudTrail attività

Nella procedura seguente viene illustrato come generare una policy per un ruolo utilizzando il AWS Management Console.

Generare una policy per un ruolo IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione sulla sinistra, scegliere Roles (Ruoli).

   > **Note**
   >
   > I passaggi per generare una policy basata sull'attività di un utente IAM sono quasi identici. A tale scopo, scegliere Users (Utenti) anziché Roles (Ruoli).

3. Nell'elenco dei ruoli dell'account, scegliere il nome del ruolo di cui si desidera utilizzare l'attività per generare una policy.
4. Nella scheda Autorizzazioni, nella sezione Genera policy basata su CloudTrail eventi, scegli Genera policy.
5. Nella pagina Genera policy, specifica il periodo di tempo in cui desideri che IAM Access Analyzer analizzi i tuoi CloudTrail eventi per verificare le azioni intraprese con il ruolo. È possibile scegliere fino a 90 giorni. Si consiglia di scegliere il periodo di tempo più breve possibile per ridurre il tempo di generazione della policy.
6. Nella sezione CloudTrail accesso, scegli un ruolo esistente adatto o crea un nuovo ruolo se non esiste un ruolo adatto. Il ruolo fornisce a IAM Access Analyzer le autorizzazioni per accedere

ai tuoi CloudTrail dati per tuo conto, per esaminare le attività di accesso e identificare i servizi e le azioni che sono stati utilizzati. Consulta Autorizzazioni richieste per generare una policy per ulteriori informazioni sulle autorizzazioni necessarie per questo ruolo.

7. Nella sezione CloudTrail Percorso da analizzare, specifica il CloudTrail percorso che registra gli eventi per l'account.

   Se scegli un CloudTrail percorso che memorizza i log in un account diverso, viene visualizzata una casella informativa sull'accesso tra account. L'accesso tra account richiede una configurazione aggiuntiva. Per ulteriori informazioni, consulta Choose a role for cross-account access più avanti in questo argomento.
8. Scegliere Generate policy (Genera policy).
9. Mentre è in corso la generazione della policy, l'utente viene rimandato alla pagina Roles (Ruoli) Summary (Riepilogo) nella scheda Permissions (Autorizzazioni). Attendere che lo stato nella sezione Policy request details (Dettagli richiesta policy) mostri Success (Operazione riuscita), quindi scegliere View generated policy (Visualizza policy generata). È possibile visualizzare la policy generata per un massimo di 7 giorni. Se si genera un'altra policy, la policy esistente viene sostituita con quella nuova generata.

Passaggio 2: Esaminare le autorizzazioni e aggiungere azioni per i servizi utilizzati

Esaminare i servizi e le azioni che Sistema di analisi degli accessi AWS IAM ha identificato come utilizzati dal ruolo. È possibile aggiungere azioni per tutti i servizi utilizzati nel modello di policy generata.

1. Leggere le seguenti sezioni:

   - Nella pagina Review permissions (Esamina le autorizzazioni), controllare l'elenco delle azioni incluse nella policy generata. Nell'elenco vengono visualizzati i servizi e le operazioni che Sistema di analisi degli accessi AWS IAM ha identificato come utilizzati dal ruolo nell'intervallo di date specificato.
   - La sezione Services used (Servizi utilizzati) mostra i servizi aggiuntivi che Sistema di analisi degli accessi AWS IAM ha identificato come utilizzati dal ruolo nell'intervallo di date specificato. Le informazioni sulle azioni utilizzate potrebbero non essere disponibili per i servizi elencati in questa sezione. Utilizzare i menu per ciascun servizio elencato per scegliere manualmente le azioni che si desidera includere nella policy.
2. Dopo avere terminato di aggiungere le azioni, scegliere Next (Avanti).

Passaggio 3: Personalizzare ulteriormente la policy generata

È possibile personalizzare ulteriormente la policy aggiungendo o rimuovendo autorizzazioni o specificando risorse.

Per personalizzare la policy generata

1. Aggiornare il modello della policy. Il modello della policy contiene i segnaposto ARN della risorsa per le azioni che supportano le autorizzazioni a livello di risorsa, come illustrato nell'immagine seguente. Il concetto di autorizzazioni a livello di risorsa indica la possibilità di specificare le risorse su cui gli utenti sono autorizzati a eseguire operazioni. Si consiglia di utilizzare gli ARN per specificare le singole risorse nella policy per le azioni che supportano le autorizzazioni a livello di risorsa. È possibile sostituire i segnaposto ARN della risorsa con ARN della risorsa validi per il caso d'uso.

   Se un'operazione non supporta le autorizzazioni a livello di risorsa, bisogna utilizzare il carattere jolly (*) per specificare che tutte le risorse possono essere interessate dall'operazione. Per scoprire quali AWS servizi supportano le autorizzazioni a livello di risorsa, consulta i servizi che funzionano con IAM.AWS Per un elenco delle operazioni in ciascun servizio e per sapere quali operazioni supportano le autorizzazioni a livello di risorsa, consultare Actions, Resources, and Condition Keys for Services AWS (Operazioni, risorse e chiavi di condizione per i servizi).

2. (Facoltativo) Aggiungere, modificare o rimuovere le istruzioni della policy JSON nel modello. Per ulteriori informazioni sulla scrittura di policy JSON, consulta Creazione di policy IAM (console).

3. Al termine della personalizzazione del modello della policy, sono disponibili le seguenti opzioni:

   - (Facoltativo) È possibile copiare la JSON nel modello da utilizzare separatamente all'esterno della pagina Generated policy (Policy generata). Ad esempio, se si desidera utilizzare la JSON per creare una policy in un account diverso. Se la policy nel modello supera il limite di 6.144 caratteri per le policy JSON, viene suddivisa in più policy.
   - Scegliere Next (Avanti) per riesaminare e creare una policy gestita nello stesso account.

Passaggio 4: Esaminare e creare una policy gestita

Se si dispone delle autorizzazioni per creare e allegare policy IAM, è possibile creare una policy gestita dalla policy generata. È quindi possibile allegare la policy a un utente o a un ruolo nel proprio account.

Per rivedere e creare una policy

1. Nella pagina Review and create managed policy (Rivedi e crea una policy gestita) digitare i valori per Name (Nome) e Description (Descrizione) (facoltativa) per la policy che si sta creando.
2. (Facoltativo) Nella sezione Summary (Riepilogo) è possibile esaminare le autorizzazioni che verranno incluse nella policy.
3. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag con IAM, consulta Tagging delle risorse IAM.
4. Al termine, effettuare una delle seguenti operazioni:

   - È possibile allegare la nuova policy direttamente al ruolo utilizzato per generare la policy. ***Per fare ciò, nella parte inferiore della pagina, seleziona la casella di controllo accanto alla politica Allega al nome. YourRole*** Quindi scegliere Create and attach policy (Crea e allega policy).
   - In caso contrario, selezionare Create policy (Crea policy). È possibile trovare la policy creata nell'elenco di policy nel riquadro di navigazione Policies (Policy) della console IAM.
5. È possibile allegare la policy creata a un'entità nel proprio account. Dopo aver collegato la policy, è possibile rimuovere tutte le altre policy di carattere troppo generale che potrebbero essere collegate all'entità. Per sapere come collegare una policy gestita, consulta Aggiunta di autorizzazioni di identità IAM (console).

Genera una politica utilizzando AWS CloudTrail i dati di un altro account

È possibile creare CloudTrail percorsi che archiviano i dati in account centrali per semplificare le attività di governo. Ad esempio, è possibile AWS Organizations creare un percorso che registri tutti gli eventi per tutti i membri dell' Account AWS organizzazione. Il percorso appartiene a un account centrale. Se desideri generare una politica per un utente o un ruolo in un account diverso da quello in cui sono archiviati i dati di CloudTrail registro, devi concedere l'accesso tra più account. Per fare ciò, sono necessarie sia una policy di ruolo che una bucket policy che conceda a IAM Access Analyzer le autorizzazioni per i log. CloudTrail Per ulteriori informazioni sulla creazione di percorsi dell'organizzazione, consulta Creazione di un percorso per un'organizzazione.

In questo esempio, supponiamo di voler generare una policy per un utente o un ruolo nell'account A. La CloudTrail traccia nell'account A memorizza CloudTrail i log in un bucket dell'account B. Prima di poter generare una policy, devi apportare i seguenti aggiornamenti:

1. Scegli un ruolo esistente o crea un nuovo ruolo di servizio che conceda a IAM Access Analyzer l'accesso al bucket dell'account B (dove sono archiviati i CloudTrail log).
2. Verifica la tua policy di proprietà degli oggetti del bucket Amazon S3 e di autorizzazioni del bucket nell'account B per consentire a Sistema di analisi degli accessi AWS IAM di accedere agli oggetti nel bucket.

Fase 1: Scelta o creazione di un ruolo per l'accesso tra account

- Nella schermata Genera policy, l'opzione Utilizza un ruolo esistente è preselezionata se nel tuo account esiste già un ruolo con le autorizzazioni richieste. In caso contrario, scegli Crea e utilizza un nuovo ruolo di servizio. Il nuovo ruolo viene utilizzato per concedere a IAM Access Analyzer l'accesso ai log nell'account B. CloudTrail

Fase 2: verifica o aggiornamento della configurazione del bucket Amazon S3 nell'account B

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco dei bucket, scegli il nome del bucket in cui sono archiviati i log dei CloudTrail percorsi.
3. Scegli la scheda Permissions (Autorizzazioni) e individua la sezione Object Ownership (Proprietà dell'oggetto).

Utilizza le impostazioni di proprietà degli oggetti del bucket Amazon S3 per controllare la proprietà dei nuovi oggetti che vengono caricati nei tuoi bucket. Per impostazione predefinita, quando altri oggetti Account AWS caricano nel tuo bucket, l'account di caricamento possiede gli oggetti. Per generare una policy, il proprietario del bucket deve possedere tutti gli oggetti all'interno del bucket. A seconda del caso d'uso dell'ACL, potrebbe essere necessario modificare l'impostazione Object Ownership (Proprietà dell'oggetto) del bucket. Imposta Object Ownership (Proprietà dell'oggetto) su una delle seguenti opzioni.

- Bucket owner enforced (Proprietario del bucket applicato) (opzione consigliata)
- Bucket owner preferred (Proprietario del bucket preferito)

> **Important**
>
> Per generare correttamente una policy, gli oggetti del bucket devono essere di proprietà del proprietario del bucket. Se scegli di utilizzare Bucket owner preferred (Proprietario del bucket preferito), puoi generare una policy solo per il periodo di tempo successivo alla modifica della proprietà dell'oggetto.

Per ulteriori informazioni sulla proprietà degli oggetti in Amazon S3, consulta la sezione Controllo della proprietà degli oggetti e disabilitazione delle ACL per il bucket nella Guida per l'utente di Amazon S3.

4. Aggiungi le autorizzazioni alla tua policy del bucket Amazon S3 nell'account B per consentire l'accesso al ruolo nell'account A.

   La policy di esempio seguente consente `ListBucket` e `GetObject` per il bucket denominato `DOC-EXAMPLE-BUCKET`. Consente l'accesso se il ruolo che accede al bucket appartiene a un account dell'organizzazione e ha un nome che inizia con `AccessAnalyzerMonitorServiceRole`. L'utilizzo di `aws:PrincipalArn` as a `Condition` nell'`Resource`elemento garantisce che il ruolo possa accedere alle attività dell'account solo se appartiene all'account A. È possibile sostituirlo `DOC-EXAMPLE-BUCKET` con il nome del bucket, `optional-prefix` con un prefisso opzionale per il bucket e `organization-id` con l'ID dell'organizzazione.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "PolicyGenerationBucketPolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/optional-prefix/AWSLogs/organization-id/
${aws:PrincipalAccount}/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "organization-id"
        },
        "StringLike": {
          "aws:PrincipalArn": "arn:aws:iam::${aws:PrincipalAccount}:role/service-
role/AccessAnalyzerMonitorServiceRole*"
        }
      }
    }
  ]
}
```

5. Se crittografi i log utilizzando AWS KMS, aggiorna la policy delle AWS KMS chiavi nell'account in cui memorizzi i CloudTrail log per concedere a IAM Access Analyzer l'accesso all'utilizzo della chiave, come illustrato nel seguente esempio di policy. Sostituisci `CROSS_ACCOUNT_ORG_TRAIL_FULL_ARN` con l'ARN per il tuo percorso e `organization-id` con l'ID dell'organizzazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
```

```
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:cloudtrail:arn":
 "CROSS_ACCOUNT_ORG_TRAIL_FULL_ARN",
          "aws:PrincipalOrgID": "organization-id"
        },
        "StringLike": {
          "kms:ViaService": [
            "access-analyzer.*.amazonaws.com",
            "s3.*.amazonaws.com"
          ]
          "aws:PrincipalArn": "arn:aws:iam::${aws:PrincipalAccount}:role/service-
role/AccessAnalyzerMonitorServiceRole*"
        }
      }
    }
  ]
}
```

Generazione di una policy basata sull' CloudTrailattività (AWS CLI)

È possibile utilizzare i seguenti comandi per generare una policy utilizzando AWS CLI.

Per generare una policy

- aws accessanalyzer start-policy-generation

Per visualizzare una policy generata

- aws access analyzer get-generated-policy

Per annullare una richiesta di generazione di policy

- aws access analyzer cancel-policy-generation

Per visualizzare un elenco di richieste di generazione di policy

- aws access analyzer list-policy-generations

Genera una politica basata sull' CloudTrailattività (API)AWS

È possibile utilizzare le seguenti operazioni per generare una politica utilizzando l' AWS API.

Per generare una policy

- StartPolicyGenerazione

Per visualizzare una policy generata

- GetGeneratedPolitica

Per annullare una richiesta di generazione di policy

- CancelPolicyGenerazione

Per visualizzare un elenco di richieste di generazione di policy

- ListPolicyGenerazioni

Servizi di generazione di policy per Sistema di analisi degli accessi IAM

La tabella seguente elenca AWS i servizi per i quali IAM Access Analyzer genera policy con informazioni a livello di azione. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi nel Service Authorization Reference.

Servizio	Prefisso del servizio
AWS Identity and Access Management Access Analyzer	access-analyzer
AWS Account Management	account
AWS Certificate Manager	acm
Flussi di lavoro gestiti da Amazon per Apache Airflow	airflow
AWS Amplify	amplify

Servizio	Prefisso del servizio
AWS Amplify Generatore di interfacce utente	amplifyuibuilder
Amazon AppIntegrations	app-integrations
AWS AppConfig	appconfig
Amazon AppFlow	appflow
AWS Application Cost Profiler	application-cost-profiler
Informazioni approfondite sulle CloudWatch applicazioni Amazon	applicationinsights
AWS App Mesh	appmesh
Amazon AppStream 2.0	appstream
AWS AppSync	appsync
Amazon Managed Service per Prometheus	aps
Amazon Athena	athena
AWS Audit Manager	auditmanager
AWS Auto Scaling	autoscaling
Marketplace AWS	aws-marketplace
AWS Backup	backup
AWS Batch	batch
Amazon Braket	braket
Budget AWS	budgets
AWS Cloud9	cloud9

Servizio	Prefisso del servizio
AWS CloudFormation	cloudformation
Amazon CloudFront	cloudfront
AWS CloudHSM	cloudhsm
Amazon CloudSearch	cloudsearch
AWS CloudTrail	cloudtrail
Amazon CloudWatch	cloudwatch
AWS CodeArtifact	codeartifact
AWS CodeDeploy	codedeploy
Amazon CodeGuru Profiler	codeguru-profiler
CodeGuru Revisore Amazon	codeguru-reviewer
AWS CodePipeline	codepipeline
AWS CodeStar	codestar
Notifiche AWS CodeStar	codestar-notifications
Amazon Cognito Identity	cognito-identity
Pool di utenti Amazon Cognito	cognito-idp
Amazon Cognito Sync	cognito-sync
Amazon Comprehend Medical	comprehendmedical
AWS Compute Optimizer	compute-optimizer
AWS Config	config

Servizio	Prefisso del servizio
Amazon Connect	connect
AWS Cost and Usage Report	cur
AWS Glue DataBrew	databrew
AWS Data Exchange	dataexchange
AWS Data Pipeline	datapipeline
DynamoDB Accelerator	dax
AWS Device Farm	devicefarm
Amazon DevOps Guru	devops-guru
AWS Direct Connect	directconnect
Amazon Data Lifecycle Manager	dlm
AWS Database Migration Service	dms
Cluster elastici Amazon DocumentDB	docdb-elastic
AWS Directory Service	ds
Amazon DynamoDB	dynamodb
Amazon Elastic Block Store	ebs
Amazon Elastic Compute Cloud	ec2
Amazon Elastic Container Registry	ecr
Amazon Elastic Container Registry Public	ecr-public
Amazon Elastic Container Service	ecs
Amazon Elastic Kubernetes Service	eks

Servizio	Prefisso del servizio
Amazon Elastic Inference	elastic-inference
Amazon ElastiCache	elasticache
AWS Elastic Beanstalk	elasticbeanstalk
Amazon Elastic File System	elasticfilesystem
Elastic Load Balancing	elasticloadbalancing
Amazon Elastic Transcoder	elastictranscoder
Amazon EMR su EKS (Containers EMR)	emr-containers
Amazon EMR Serverless	emr-serverless
OpenSearch Servizio Amazon	es
Amazon EventBridge	events
Amazon CloudWatch evidentemente	evidently
Amazon FinSpace	finspace
Amazon Data Firehose	firehose
AWS Fault Injection Service	fis
AWS Firewall Manager	fms
Amazon Fraud Detector	frauddetector
Amazon FSx	fsx
Amazon GameLift	gamelift
Servizio di posizione Amazon	geo

Servizio	Prefisso del servizio
Amazon S3 Glacier	glacier
Grafana gestito da Amazon	grafana
AWS IoT Greengrass	greengrass
AWS Ground Station	groundstation
Amazon GuardDuty	guardduty
AWS HealthLake	healthlake
Amazon Honeycode	honeycode
AWS Identity and Access Management	iam
AWS Archivio di identità	identitystore
EC2 Image Builder	imagebuilder
Amazon Inspector Classic	inspector
Amazon Inspector	inspector2
AWS IoT	iot
AWS IoT Analytics	iotanalytics
AWS IoT Core Device Advisor	iotdeviceadvisor
AWS IoT Events	iotevents
AWS IoT Fleet Hub	iotfleethub
AWS IoT SiteWise	iotsitewise
AWS IoT TwinMaker	iottwinmaker
AWS IoT Wireless	iotwireless

Servizio	Prefisso del servizio
Amazon Interactive Video Service	ivs
Amazon Interactive Video Service Chat	ivschat
Amazon Managed Streaming per Apache Kafka	kafka
Amazon Managed Streaming per Kafka Connect	kafkaconnect
Amazon Kendra	kendra
Amazon Kinesis	kinesis
Amazon Kinesis Analytics V2	kinesisanalytics
AWS Key Management Service	kms
AWS Lambda	lambda
Amazon Lex	lex
AWS License Manager Gestore abbonamenti Linux	license-manager-linux-subscriptions
Amazon Lightsail	lightsail
CloudWatch Registri Amazon	log
Amazon Lookout per le apparecchiature	lookoutequipment
Amazon Lookout per le metriche	lookoutmetrics
Amazon Lookout per Vision	lookoutvision
AWS Mainframe Modernization	m2
Blockchain gestita da Amazon	managedblockchain
AWS Elemental MediaConnect	mediaconnect

Servizio	Prefisso del servizio
AWS Elemental MediaConvert	mediaconvert
AWS Elemental MediaLive	medialive
AWS Elemental MediaStore	mediastore
AWS Elemental MediaTailor	mediatailor
Amazon MemoryDB per Redis	memorydb
AWS Application Migration Service	mgn
AWS Migration Hub	mgh
Suggerimenti AWS sulla strategia di Migration Hub	migration hub-strategy
Amazon Pinpoint	mobiletargeting
Amazon MQ	mq
AWS Network Manager	networkmanager
Amazon Nimble Studio	nimble
AWS HealthOmics	omics
AWS OpsWorks	opsworks
AWS OpsWorks CM	opsworks-cm
AWS Outposts	outposts
AWS Organizations	organizations
AWS Panorama	panorama
AWS Approfondimenti sulle prestazioni	pi

Servizio	Prefisso del servizio
EventBridge Tubi Amazon	pipes
Amazon Polly	polly
Profili cliente Amazon Connect	profilo
Amazon QLDB	qldb
AWS Resource Access Manager	ram
AWS Cestino di riciclaggio	rbin
Amazon Relational Database Service	rds
Amazon Redshift	redshift
API dati di Amazon Redshift	redshift-data
AWS Migration Hub Refactor Spaces	refactor-spaces
Amazon Rekognition	rekognition
AWS Resilience Hub	resiliencehub
Esploratore di risorse AWS	resource-explorer-2
AWS Resource Groups	resource-groups
AWS RoboMaker	robomaker
AWS Identity and Access Management Ruoli ovunque	rolesanywhere
Amazon Route 53	route53
Controlli di ripristino Amazon Route 53	route53-recovery-control-config

Servizio	Prefisso del servizio
Preparazione al ripristino di Amazon Route 53	route53-recovery-readiness
Amazon Route 53 Resolver	route53resolver
AWS CloudWatch RUM	rum
Amazon Simple Storage Service	s3
Amazon S3 su Outposts	s3-outposts
Funzionalità SageMaker geospaziali di Amazon	sagemaker-geospatial
Savings Plans	savingsplans
EventBridge Schemi Amazon	schemas
Amazon SimpleDB	sdb
AWS Secrets Manager	secretsmanager
AWS Security Hub	securityhub
Amazon Security Lake	securitylake
AWS Serverless Application Repository	serverlessrepo
AWS Service Catalog	servicecatalog
AWS Cloud Map	servicediscovery
Service Quotas	servicequotas
Amazon Simple Email Service	ses
AWS Shield	shield
AWS Signer	signer

Servizio	Prefisso del servizio
AWS SimSpace Weaver	simspaceweaver
AWS Server Migration Service	sms
Servizio di SMS e messaggi vocali Amazon Pinpoint	sms-voice
AWS Snowball	snowball
Amazon Simple Queue Service	sqs
AWS Systems Manager	ssm
AWS Systems Manager Incident Manager	ssm-incidents
AWS Systems Manager per SAP	ssm-sap
AWS Step Functions	states
AWS Security Token Service	sts
Amazon Simple Workflow Service	swf
Amazon CloudWatch Synthetics	synthetics
AWS Resource Groups Tagging API	tag
Amazon Textract	textract
Amazon Timestream	timestream
AWS Costruttore di reti di telecomunicazioni	tnb
Amazon Transcribe	transcribe
AWS Transfer Family	transfer
Amazon Translate	translate
Amazon Connect Voice ID	voiceid

Servizio	Prefisso del servizio
Amazon VPC Lattice	vpc-lattice
AWS WAFV2	wafv2
AWS Well-Architected Tool	wellarchitected
Amazon Connect Wisdom	wisdom
Amazon WorkLink	worklink
Amazon WorkSpaces	workspace
AWS X-Ray	xray

Quote Sistema di analisi degli accessi AWS IAM

Sistema di analisi degli accessi AWS IAM ha le seguenti quote:

Risorsa	Quota predefinita	Quota massima
Numero massimo di analizzatori a livello di account per tipo di analizzatore dell' Account AWS per regione	1	1
Numero massimo di analizzatori a livello di organizzazione per tipo di analizzatore dell' Account AWS per regione	5	20[1]
Numero massimo di regole di archiviazione per analizzatore	100 Ogni regola di archivio può avere fino a 20 valori per criterio.	1.000[1]

Risorsa	Quota predefinita	Quota massima
Numero massimo di anteprime di accesso per analizzatore all'ora	1.000	1.000
AWS CloudTrail file di registro elaborati per generazioni di policy	100.000	100.000
Generazioni simultanee di policy	1	1
Dimensione dei AWS CloudTrail dati di generazione delle politiche	25 GB	25 GB
AWS CloudTrail Intervallo di tempo di generazione delle politiche	90 giorni	90 giorni
Generazioni di policy al giorno	Africa (Città del Capo): 5 Asia Pacifico (Hong Kong): 5 Europa (Milano): 5 Medio Oriente (Bahrein): 5 Tutte le altre regioni supportate: 50 **Note** Le richieste di generazione delle policy annullate si applicano alla quota giornaliera.	Africa (Città del Capo): 5 Asia Pacifico (Hong Kong): 5 Europa (Milano): 5 Medio Oriente (Bahrein): 5 Tutte le altre regioni supportate: 50

[1]Alcune quote possono essere configurate dal cliente tramite Service Quotas.

Risoluzione dei problemi di IAM

Se si verificano problemi di accesso rifiutato o problemi simili durante l'utilizzo di AWS Identity and Access Management (IAM), consulta gli argomenti in questa sezione.

Argomenti

Risoluzione dei problemi IAM generali

Utilizza le informazioni qui riportate per eseguire la diagnosi e risolvere problemi comuni durante l'utilizzo di AWS Identity and Access Management (IAM).

Problemi

Non riesco ad accedere al mio account AWS

Verifica di disporre delle credenziali corrette e di utilizzare il metodo corretto per accedere. Per ulteriori informazioni, consulta Risoluzione dei problemi di accesso nella Guida per l'utente di Accedi ad AWS .

Chiavi di accesso smarrite

Le chiavi di accesso sono costituite da due parti:

- Identificatore della chiave di accesso: Questo non è un segreto e può essere visualizzato nella console IAM ovunque le chiavi di accesso siano elencate, ad esempio nella pagina di riepilogo dell'utente.
- Chiave di accesso segreta: questa informazione viene fornita quando si crea inizialmente la coppia di chiavi di accesso. Proprio come una password, non può essere recuperata in seguito. Se la chiave di accesso segreta viene persa, è necessario creare una nuova coppia di chiavi di accesso. Se si disponi già del numero massimo di chiavi di accesso, è necessario eliminare una coppia esistente prima di crearne un'altra.

Per ulteriori informazioni, consulta Reimpostazione delle password o delle chiavi di accesso perse o dimenticate per AWS.

Variabili della policy non funzionanti

- Verificare che tutte le policy che includono variabili includano il seguente numero di versione nella policy: `"Version": "2012-10-17"`. Senza il numero di versione corretto, le variabili non vengono sostituite durante la valutazione. Al contrario, le variabili vengono valutate letteralmente. Qualsiasi policy che non include variabili continuerà a funzionare se si include il numero di versione più recente.

  Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called “Controllo delle versioni delle policy IAM”.

- Verificare che le variabili della policy applichino la distinzione maiuscole/minuscole corretta. Per informazioni dettagliate, vedi Elementi delle policy IAM: variabili e tag.

Le modifiche che apporto non sono sempre immediatamente visibili

Essendo un servizio a cui si accede da computer in data center presenti in tutto il mondo, IAM utilizza un modello di elaborazione distribuito denominato consistenza finale. Qualsiasi modifica apportata a IAM (o ad altri AWS servizi), compresi i tag utilizzati nel controllo degli accessi basato sugli attributi (ABAC), richiede tempo per diventare visibile da tutti gli endpoint possibili. Alcuni dei ritardi sono dovuti al tempo necessario per inviare i dati da un server a un altro, da una zona di replica a un'altra e da una regione a un'altra nel mondo. IAM utilizza inoltre la memorizzazione nella cache per migliorare le prestazioni, è possibile che ciò aumenti ulteriormente il tempo richiesto, in quanto la modifica potrebbe risultare visibile solo dopo il timeout dei dati memorizzati nella cache.

È necessario progettare le applicazioni globali in modo da considerare questi potenziali ritardi e assicurarsi che funzionino come previsto, anche quando una modifica apportata in una posizione non è immediatamente visibile in un'altra. Tali modifiche includono la creazione o l'aggiornamento di utenti, gruppi, ruoli, o policy. Si consiglia di non includere tali modifiche IAM nei percorsi critici e ad alta disponibilità del codice dell'applicazione. Al contrario, apporta modifiche IAM in un'inizializzazione separata o in una routine di configurazione che si esegue meno frequentemente. Inoltre, assicurarsi di verificare che le modifiche siano state propagate prima che i flussi di lavoro di produzione dipendano da esse.

Per ulteriori informazioni su come alcuni altri AWS servizi ne risentono, consulta le seguenti risorse:

- Amazon DynamoDB: Qual è il modello di consistenza di Amazon DynamoDB? in Domande frequenti su DynamoDB e Consistenza di lettura nella Guida per gli sviluppatori di Amazon DynamoDB.
- Amazon EC2: Consistenza finale di EC2 nella Documentazione di riferimento dell'API Amazon EC2.
- Amazon EMR: garantire la coerenza nell'utilizzo di Amazon S3 e MapReduce Amazon Elastic for ETL per i flussi di lavoro AWS nel blog sui Big Data
- Amazon Redshift: Gestione della consistenza dei dati nella Guida per gli sviluppatori di Amazon Redshift Database
- Amazon S3: Modello di consistenza dei dati di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service

Non sono autorizzato a eseguire: iam: MFADevice DeleteVirtual

Quando si tenta di assegnare o rimuovere un dispositivo MFA virtuale per sé stessi o altri, è possibile che venga visualizzato il seguente errore:

```
User: arn:aws:iam::123456789012:user/Diego is not authorized to perform:
 iam:DeleteVirtualMFADevice on resource: arn:aws:iam::123456789012:mfa/Diego with an
 explicit deny
```

Ciò può accadere se qualcuno in precedenza ha iniziato ad assegnare un dispositivo MFA virtuale a un utente nella console IAM e poi ha annullato il processo. Questa operazione crea un dispositivo MFA virtuale per l'utente in IAM, ma non lo assegna mai all'utente. È necessario eliminare il dispositivo MFA virtuale esistente prima di poter creare un nuovo dispositivo MFA virtuale con lo stesso nome del dispositivo.

Per risolvere questo problema, un amministratore non dovrebbe modificare le policy di autorizzazioni. L'amministratore deve invece utilizzare l' AWS API AWS CLI o per eliminare il dispositivo MFA virtuale esistente ma non assegnato.

Per eliminare un dispositivo MFA virtuale esistente, ma non assegnato

1. Visualizzare i dispositivi MFA virtuali nel proprio account.
   - AWS CLI: `aws iam list-virtual-mfa-devices`
   - AWS API: `ListVirtualMFADevices`
2. Nella risposta, individuare l'ARN del dispositivo MFA virtuale dell'utente per il quale si sta tentando di correggere l'anomalia.
3. Eliminare il dispositivo MFA virtuale.
   - AWS CLI: `aws iam delete-virtual-mfa-device`
   - AWS API: `DeleteVirtualMFADevice`

Come posso creare utenti IAM in modo sicuro?

Se hai dipendenti che richiedono l'accesso a AWS, puoi scegliere di creare utenti IAM o utilizzare IAM Identity Center per l'autenticazione. Se utilizzi IAM, ti AWS consiglia di creare un utente IAM e di comunicare in modo sicuro le credenziali al dipendente. Se non ci si trova fisicamente accanto

al dipendente, si consiglia di utilizzare un flusso di lavoro sicuro per comunicare le credenziali ai dipendenti.

Utilizza il seguente flusso di lavoro per creare in modo sicuro un nuovo utente in IAM:

1. Crea un nuovo utente utilizzando la AWS Management Console. Scegli di concedere AWS Management Console l'accesso con una password generata automaticamente. Se necessario, seleziona la casella di controllo accanto a L'utente deve creare una nuova password all'accesso successivo. Non aggiungere una policy di autorizzazione all'utente fino a quando non ha cambiato la password.
2. Dopo avere aggiunto l'utente, copia l'URL di accesso, il nome utente e la password per il nuovo utente. Per visualizzare la password, scegli Mostra.
3. Invia la password al tuo dipendente utilizzando un metodo di comunicazione sicuro della tua azienda, ad esempio e-mail, chat o un sistema di ticket. Separatamente, fornisci agli utenti il collegamento alla console utente IAM e il relativo nome utente. Chiedi al dipendente di confermare che riesce ad accedere correttamente prima di concedergli le autorizzazioni.
4. Dopo che il dipendente ha confermato, aggiungi le autorizzazioni necessarie. Come buona prassi di sicurezza, aggiungi una policy che richiede all'utente di autenticarsi utilizzando la MFA per gestire le proprie credenziali. Per un esempio di policy, consulta AWS: consente agli utenti IAM autenticati tramite MFA di gestire le proprie credenziali nella pagina Credenziali di sicurezza.

Risorse aggiuntive

Le seguenti risorse possono aiutarti a risolvere i problemi mentre lavori con. AWS

- AWS CloudTrail Guida per l'utente: consente AWS CloudTrail di tenere traccia di una cronologia delle chiamate API effettuate AWS e archiviare tali informazioni nei file di registro. Ciò consente di determinare quali utenti e account hanno effettuato l'accesso alle risorse nell'account, quando sono state effettuate le chiamate, quali operazioni sono state richieste e altro ancora. Per ulteriori informazioni, consulta Registrazione delle chiamate IAM e AWS STS API con AWS CloudTrail.
- AWS Knowledge Center: trova domande frequenti e collegamenti ad altre risorse per aiutarti a risolvere i problemi.
- AWS Centro assistenza: ottieni supporto tecnico.
- AWS Premium Support Center: ottieni supporto tecnico premium.

Risoluzione dei problemi dei messaggi di errore di accesso rifiutato

Gli errori di accesso negato vengono visualizzati quando si nega AWS in modo esplicito o implicito una richiesta di autorizzazione. Una negazione esplicita si verifica quando una politica contiene una `Deny` dichiarazione per l'azione specifica. AWS Un diniego implicito si verifica quando non è presente un'istruzione `Deny` applicabile e non è presente neppure un'istruzione `Allow` applicabile. Dato che una policy IAM nega un principale IAM per impostazione predefinita, la policy deve consentire esplicitamente al principale di eseguire un'operazione. In caso contrario, la policy nega implicitamente l'accesso. Per ulteriori informazioni, consulta Differenza tra rifiuto esplicito e implicito.

Se più policy dello stesso tipo negano una richiesta di autorizzazione, AWS non specifica il numero di policy nel messaggio di errore di accesso negato. Se più tipi di policy negano una richiesta di autorizzazione, AWS include solo uno di questi tipi di policy nel messaggio di errore.

> **Important**
>
> Hai problemi ad accedere a AWS? Assicurati di essere nella pagina di accesso AWS corretta per il tuo tipo di utente. Se sei il Utente root dell'account AWS (proprietario dell'account), puoi accedere AWS utilizzando le credenziali che hai configurato quando hai creato il Account AWS. Se sei un utente IAM, l'amministratore dell'account può fornirti le credenziali che puoi utilizzare per accedere ad AWS. Se hai bisogno di richiedere assistenza, non utilizzare il link di feedback in questa pagina, poiché il modulo non AWS Support viene ricevuto dal team addetto alla AWS documentazione. Invece, nella pagina Contattaci scegli Ancora impossibile accedere al tuo AWS account, quindi scegli una delle opzioni di supporto disponibili.

Ricevo un messaggio di «accesso negato» quando faccio una richiesta a un AWS servizio

- Controlla se il messaggio di errore include il tipo di policy responsabile del rifiuto dell'accesso. Ad esempio, se l'errore indica che l'accesso è stato negato a causa di una policy di controllo dei servizi (SCP), puoi concentrarti sulla risoluzione dei problemi SCP. Se conosci il tipo di policy, puoi anche verificare la presenza di un'istruzione di negazione o di un'autorizzazione mancante per l'azione specifica nelle policy di tale tipo di policy. Se il messaggio di errore non riporta il tipo di policy responsabile del rifiuto dell'accesso, utilizza le altre linee guida in questa sezione per risolvere i problemi.

- Verificare di disporre dell'autorizzazione della policy basata su identità necessaria per chiamare l'operazione e le risorse richieste. Se sono impostate delle condizioni, è necessario soddisfare anche tali condizioni quando si invia la richiesta. Per informazioni sulla visualizzazione o la modifica delle policy IAM per un utente, gruppo o ruolo, consulta Gestione di policy IAM.
- Se AWS Management Console restituisce un messaggio che indica che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore ti ha fornito le credenziali di accesso o il link di accesso.

  L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `widgets:`*`GetWidget`* fittizie.

  ```
  User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
   widgets:GetWidget on resource: my-example-widget
  ```

  In questo caso, Mateo deve richiedere al suo amministratore di aggiornare le sue policy per poter accedere alla risorsa *`my-example-widget`* utilizzando l'operazione `widgets:`*`GetWidget`*.
- Stai cercando di accedere a un servizio che supportapolicy basate sulle risorse, ad esempio Amazon S3, Amazon SNS o Amazon SQS? In tal caso, verificare che la policy specifichi l'utente come principale capitale e conceda l'accesso. Se si effettua una richiesta a un servizio all'interno dell'account, le policy basate su identità o le policy basate su risorse possono concedere l'autorizzazione. Se si effettua una richiesta a un servizio in un altro account, sia le policy basate su identità che le policy basate su risorse devono concedere l'autorizzazione. Per scoprire quali servizi supportano le policy basate su risorse, consultare la pagina AWS servizi che funzionano con IAM.
- Se la policy include una condizione con una coppia chiave-valore, esaminala con attenzione. Gli esempi includono la chiave di condizione `aws:RequestTag/`*`tag-key`*globale AWS KMS `kms:EncryptionContext:`*`encryption_context_key`*, la e la chiave di `ResourceTag/`*`tag-key`* condizione supportata da più servizi. Verifica che il nome della chiave non corrisponda a più risultati. Poiché i nomi delle chiavi di distinzioni non fanno distinzione tra maiuscole e minuscole, una condizione che verifica la presenza di una chiave denominata `foo` corrisponde a `foo`, `Foo` o `F00`. Se la richiesta include più coppie chiave-valore con nomi di chiavi che cambiano solo la dimensione dei caratteri, l'accesso potrebbe essere inaspettatamente negato. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Condition.
- Se è presente un limite delle autorizzazioni, è necessario verificare che la policy utilizzata per il limite delle autorizzazioni consenta la richiesta. Se le policy basate su identità consentono la richiesta, ma il limite delle autorizzazioni non la consente, la richiesta viene rifiutata. Il limite

delle autorizzazioni controlla il numero massimo di autorizzazioni che è possibile concedere a un'identità principale IAM (utente o ruolo). Le policy basate su risorse non sono limitate dai limiti delle autorizzazioni. I limiti delle autorizzazioni non sono comuni. Per ulteriori informazioni su come AWS valuta le politiche, vedereLogica di valutazione delle policy.

- Se stai firmando manualmente richieste API (senza utilizzare gli SDK AWS), verifica di aver firmato correttamente la richiesta.

Messaggio di accesso rifiutato quando si effettua una richiesta con credenziali di sicurezza temporanee

- Innanzitutto, occorre verificare che l'accesso non venga negato per un motivo non legato alle credenziali temporanee. Per ulteriori informazioni, consulta Ricevo un messaggio di «accesso negato» quando faccio una richiesta a un AWS servizio.
- Verificare che il servizio accetti le credenziali di sicurezza temporanee, consultare AWS servizi che funzionano con IAM.
- Verifica che le tue richieste vengano firmate correttamente e che il formato della richiesta sia valido. Per ulteriori informazioni, consulta la documentazione del kit di strumenti o Utilizzo di credenziali temporanee con le risorse AWS.
- Verifica che le credenziali di sicurezza provvisorie non siano scadute. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM.
- Verifica che l'utente o il ruolo IAM dispongano delle autorizzazioni corrette. Le autorizzazioni per le credenziali di sicurezza temporanee sono derivate da un utente o ruolo IAM. Di conseguenza, le autorizzazioni sono limitate a quelle che vengono concesse al ruolo di cui hai assunto le credenziali temporanee. Per Scopri come vengono determinate le autorizzazioni per le credenziali di sicurezza temporanee, consultare Controllo delle autorizzazioni per le credenziali di sicurezza temporanee.
- Se hai assunto un ruolo, la sessione del ruolo potrebbe essere limitata da policy di sessione. Quando richiedi credenziali di sicurezza temporanee a livello di codice AWS STS, puoi facoltativamente passare policy di sessione in linea o gestite. Le policy di sessione sono policy avanzate che vengono passate come parametro durante la creazione di una sessione temporanea per un ruolo a livello di programmazione. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. In alternativa, se l'amministratore o un programma personalizzato fornisce le credenziali temporanee, potrebbero includere policy di sessione per limitare l'accesso.

- Se sei un utente federato, la tua sessione potrebbe essere limitata da policy di sessione. Diventi un utente federato accedendo AWS come utente IAM e quindi richiedendo un token di federazione. Per ulteriori informazioni sugli utenti federati, consulta GetFederationToken: federazione tramite un gestore identità personalizzato. Se l'utente o un gestore identità ha passato policy di sessione durante la richiesta di un token di federazione, la sessione è limitata da quelle policy. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate sull'identità dell'utente IAM e delle policy di sessione. Per ulteriori informazioni sulle policy di sessione, consulta Policy di sessione.
- Se si accede a una risorsa che dispone di una policy basata sulle risorse tramite un ruolo, verificare che la policy conceda le autorizzazioni per il ruolo. Ad esempio, la policy seguente permette `MyRole` dell'account `111122223333` per l'accesso a `MyBucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "S3BucketPolicy",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::111122223333:role/MyRole"]},
    "Action": ["s3:PutObject"],
    "Resource": ["arn:aws:s3:::MyBucket/*"]
  }]
}
```

Esempi di messaggi di errore di accesso negato

Per la maggior parte, i messaggi di errore di accesso negato sono visualizzati nel formato `User` *`user`* `is not authorized to perform` *`action`* `on` *`resource`* `because` *`context`*. In questo esempio, *`user`* (utente) è il nome della risorsa Amazon (ARN) che non riceve l'accesso, *`action`* (operazione) è l'operazione di servizio che la policy nega e *`resource`* (risorsa) è l'ARN della risorsa su cui la policy ha effetto. Il campo *`context`* (contesto) rappresenta il contesto aggiuntivo sul tipo di policy che illustra perché l'accesso è negato.

Quando una policy nega esplicitamente l'accesso perché contiene un'`Deny`istruzione, AWS include la frase `with an explicit deny in a` *`type`* `policy` nel messaggio di errore di accesso negato. Quando la policy nega implicitamente l'accesso, AWS include la frase `because no` *`type`* `policy allows the` *`action`* `action` nel messaggio di errore di accesso negato.

> **Note**
>
> Alcuni AWS servizi non supportano questo formato di messaggio di errore di accesso negato. Il contenuto dei messaggi di errore di accesso negato può variare a seconda del servizio che effettua la richiesta di autorizzazione.

Gli esempi seguenti mostrano il formato di vari tipi di messaggi di errore di accesso negato.

Accesso negato a causa di una policy di controllo dei servizi: diniego implicito

1. Verifica un'istruzione `Allow` mancante relativa all'operazione nelle tue policy di controllo dei servizi (SCP). Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua SCP aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

```
User: arn:aws:iam::777788889999:user/JohnDoe is not authorized to perform:
codecommit:ListRepositories because no service control policy allows the
 codecommit:ListRespositories action
```

Accesso negato a causa di una policy di controllo dei servizi: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` relativa all'operazione nelle tue policy di controllo dei servizi (SCP). Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua SCP rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS Organizations .

```
User: arn:aws:iam::777788889999:user/JohnDoe is not authorized to perform:
codecommit:ListRepositories with an explicit deny in a service control policy
```

Accesso negato a causa di una policy dell'endpoint VPC: diniego implicito

1. Verifica l'assenza di un'istruzione `Allow` per l'azione nelle tue policy dell'endpoint del cloud privato virtuale (VPC). Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.

2. Aggiorna la tua policy sugli endpoint VPC aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta Aggiornamento di una policy dell'endpoint VPC nella AWS PrivateLink Guida.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codecommit:ListRepositories because no VPC endpoint policy allows the
 codecommit:ListRepositories action
```

Accesso negato a causa di una policy dell'endpoint VPC: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nelle tue policy dell'endpoint del cloud privato virtuale (VPC). Per l'esempio seguente, l'operazione è `codedeploy:ListDeployments`.
2. Aggiorna la tua policy sugli endpoint VPC rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Aggiornamento di una policy dell'endpoint VPC nella AWS PrivateLink Guida.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codedeploy:ListDeployments on resource: arn:aws:codedeploy:us-
east-1:123456789012:deploymentgroup:* with an explicit deny in a VPC endpoint policy
```

Accesso negato a causa di limiti delle autorizzazioni: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nel limite delle autorizzazioni. Per l'esempio seguente, l'operazione è `codedeploy:ListDeployments`.
2. Aggiorna il limite delle autorizzazioni aggiungendo l'istruzione `Allow`relativa alla tua policy IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codedeploy:ListDeployments on resource: arn:aws:codedeploy:us-
east-1:123456789012:deploymentgroup:* because no permissions boundary allows the
 codedeploy:ListDeployments action
```

Accesso negato a causa di un limite delle autorizzazioni: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nel limite delle autorizzazioni. Per l'esempio seguente, l'operazione è `sagemaker:ListModels`.
2. Aggiorna il limite delle autorizzazioni rimuovendo l'istruzione `Deny`relativa alla tua policy IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM e Modifica delle policy IAM.

```
User: arn:aws:iam::777788889999:user/JohnDoe is not authorized to perform:
sagemaker:ListModels with an explicit deny in a permissions boundary
```

Accesso negato a causa di policy di sessione: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nelle tue policy di sessione. Per l'esempio seguente, l'operazione è `codecommit:ListRepositories`.
2. Aggiorna la tua policy di sessione aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta la sezione Policy di sessione e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codecommit:ListRepositories because no session policy allows the
 codecommit:ListRepositories action
```

Accesso negato a causa di policy di sessione: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nelle tue policy di sessione. Per l'esempio seguente, l'operazione è `codedeploy:ListDeployments`.
2. Aggiorna la tua policy di sessione rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta la sezione Policy di sessione e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codedeploy:ListDeployments on resource: arn:aws:codedeploy:us-
east-1:123456789012:deploymentgroup:* with an explicit deny in a sessions policy
```

Accesso negato a causa di policy basate sulle risorse: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nella tua policy basata sulle risorse. Per l'esempio seguente, l'operazione è `secretsmanager:GetSecretValue`.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta la sezione Policy basate sulle risorse e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
secretsmanager:GetSecretValue because no resource-based policy allows the
 secretsmanager:GetSecretValue action
```

Accesso negato a causa di policy basate sulle risorse: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nella tua policy basata sulle risorse. Per l'esempio seguente, l'operazione è `secretsmanager:GetSecretValue`.
2. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta la sezione Policy basate sulle risorse e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
secretsmanager:GetSecretValue on resource: arn:aws:secretsmanager:us-
east-1:123456789012:secret:* with an explicit deny in a resource-based policy
```

Accesso negato a causa di policy di attendibilità dei ruoli: diniego implicito

1. Verifica la presenza di un'istruzione `Allow` mancante relativa all'azione nella tua policy di attendibilità dei ruoli. Per l'esempio seguente, l'operazione è `sts:AssumeRole`.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta la sezione Policy basate sulle risorse e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
sts:AssumeRole because no role trust policy allows the sts:AssumeRole action
```

Accesso negato a causa di policy di attendibilità dei ruoli: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nella tua policy di attendibilità dei ruoli. Per l'esempio seguente, l'operazione è `sts:AssumeRole`.
2. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta la sezione Policy basate sulle risorse e Modifica delle policy IAM.

```
User: arn:aws:iam::777788889999:user/JohnDoe is not authorized to perform:
sts:AssumeRole with an explicit deny in the role trust policy
```

Accesso negato a causa di policy basate sull'identità: diniego implicito

1. Verifica l'eventuale mancanza di un'istruzione `Allow` per l'azione nelle politicy basate sull'identità collegate all'identità. Nel seguente esempio, l'azione `codecommit:ListRepositories` è collegata all'utente `JohnDoe`.
2. Aggiorna la tua policy aggiungendo l'istruzione `Allow`. Per ulteriori informazioni, consulta la sezione Policy basate sull'identità e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codecommit:ListRepositories because no identity-based policy allows the
 codecommit:ListRepositories action
```

Accesso negato a causa di policy basate sull'identità: diniego esplicito

1. Verifica la presenza di un'istruzione `Deny` esplicita per l'azione nelle politicy basate sull'identità collegate all'identità. Nel seguente esempio, l'azione è `codedeploy:ListDeployments` collegata all'utente`JohnDoe`.
2. Aggiorna la tua policy rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta la sezione Policy basate sull'identità e Modifica delle policy IAM.

```
User: arn:aws:iam::123456789012:user/JohnDoe is not authorized to perform:
codedeploy:ListDeployments on resource: arn:aws:codedeploy:us-
east-1:123456789012:deploymentgroup:* with an explicit deny in an identity-based policy
```

Accesso negato quando una richiesta VPC ha esito negativo a causa di un'altra policy

1. Verifica la presenza di un'istruzione `Deny` esplicita relativa all'azione nelle tue policy di controllo dei servizi (SCP). Per l'esempio seguente, l'operazione è `SNS:Publish`.
2. Aggiorna la tua SCP rimuovendo l'istruzione `Deny`. Per ulteriori informazioni, consulta Aggiornamento di una SCP nella Guida per l'utente di AWS IAM Identity Center .

```
User: arn:aws:sts::111122223333:assumed-role/role-name/role-session-name is not
 authorized to perform:
SNS:Publish on resource: arn:aws:sns:us-east-1:444455556666:role-name-2
with an explicit deny in a VPC endpoint policy transitively through a service control
 policy
```

Risoluzione dei problemi relativi alle policy IAM

Una policy è un'entità AWS che, se associata a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale, ad esempio un utente, effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. Le politiche vengono archiviate AWS come documenti JSON allegati ai principali come politiche basate sull'identità o alle risorse come politiche basate sulle risorse. È possibile collegare una policy basata sull'identità a un principale (o identità), ad esempio un gruppo, un utente o un ruolo IAM. Le policy basate sulle identità includono policy gestite da AWS , policy gestite dal cliente e policy inline. È possibile creare e modificare le politiche gestite dai clienti utilizzando le opzioni di editor Visual e JSON AWS Management Console . Quando si visualizza una politica in AWS Management Console, è possibile visualizzare un riepilogo delle autorizzazioni concesse da tale politica. L'editor visivo e i riepiloghi di policy consentono di individuare e risolvere errori comuni durante la gestione delle policy IAM.

Tieni presente che tutte le policy IAM vengono archiviate utilizzando una sintassi che inizia con le regole di JavaScript Object Notation (JSON). Non è necessario conoscere questa sintassi per creare o gestire le policy. È possibile creare e modificare una policy utilizzando l'editor visivo nella AWS Management Console. Per ulteriori informazioni sulla sintassi JSON nelle policy IAM, consulta Sintassi del linguaggio della policy JSON IAM .

Risoluzione dei problemi degli argomenti della policy IAM

- Risoluzione dei problemi tramite l'editor visivo

Risoluzione dei problemi tramite l'editor visivo

Quando si crea o si modifica una policy gestita dal cliente, è possibile utilizzare le informazioni nell'editor Visivo per semplificare la risoluzione degli errori della policy. Per visualizzare un esempio dell'editor visivo per creare una policy, consultare the section called “Controllo dell'accesso alle identità”.

Modifica della struttura delle policy

Quando crei una policy, la AWS convalida, la elabora e la trasforma prima di archiviarla. Quando AWS restituisce la policy in risposta a una richiesta dell'utente o la visualizza nella console, la AWS

trasforma nuovamente in un formato leggibile dall'uomo senza modificare le autorizzazioni concesse dalla policy. Questo può causare differenze in ciò che viene visualizzato nell'editor visivo o nella scheda JSON: i blocchi di autorizzazioni dell'editor visivo possono essere aggiunti, rimossi o riordinati e il contenuto all'interno di un blocco può essere ottimizzato. Nella scheda JSON lo spazio bianco non significativo può essere rimosso e gli elementi all'interno di mappe JSON possono essere riordinati. Inoltre, gli Account AWS ID all'interno degli elementi principali possono essere sostituiti dall'ARN di. Utente root dell'account AWS A causa di queste possibili modifiche, non è possibile confrontare i documenti di policy JSON come stringhe.

Quando crei una policy gestita dai clienti in AWS Management Console, puoi scegliere di lavorare interamente nell'editor JSON. Se non si apportano modifiche nell'editor Visivo e si seleziona Successivo dall'editor JSON, è meno probabile che la policy venga ristrutturata. Tuttavia, se si crea una policy e si utilizza l'editor Visivo per apportare le eventuali modifiche oppure se scegli Successivo dall'opzione dell'editor Visivo, IAM potrebbe ristrutturare la policy per ottimizzarne l'aspetto nell'editor visivo.

Questa ristrutturazione esiste solo nella sessione di modifica e non viene salvata automaticamente.

Se la policy è stata ristrutturata nella sessione di modifica, IAM determina se salvare la ristrutturazione in base alle seguenti situazioni:

Utilizzo di questa opzione dell'editor	Se si modifica la policy	Quindi scegli Successivo da questa scheda	Quando si sceglie Save changes (Salva modifiche)
Visivo	Modificata	Visivo	La policy viene ristrutturata
Visivo	Modificata	JSON	La policy viene ristrutturata
Visivo	Non modificata	Visivo	La policy viene ristrutturata
JSON	Modificata	Visivo	La policy viene ristrutturata

Utilizzo di questa opzione dell'editor	Se si modifica la policy	Quindi scegli Successivo da questa scheda	Quando si sceglie Save changes (Salva modifiche)
JSON	Modificata	JSON	La struttura della policy non viene modificata
JSON	Non modificata	JSON	La struttura della policy non viene modificata

IAM potrebbe ristrutturare policy complesse o policy che hanno blocchi di autorizzazione o istruzioni per permettere più servizi, tipi di risorse o chiavi di condizioni.

Scelta di un ARN della risorsa nell'editor visivo

Quando si crea o si modifica una policy utilizzando l'editor visivo, è necessario prima selezionare un servizio, quindi selezionare operazioni da quel servizio. Se il servizio e le operazioni selezionate supportano la scelta di risorse specifiche, l'editor visivo elenca i tipi di risorse supportati. È quindi possibile selezionare Add ARN (Aggiungi ARN) per fornire i dettagli sulla risorsa. È possibile selezionare tra le seguenti opzioni per aggiungere un ARN per un tipo di risorsa.

- Utilizza il builder di ARN: in base al tipo di risorsa, è possibile che siano visualizzati campi diversi per la creazione dell'ARN. È anche possibile selezionare Any (Qualsiasi) per fornire le autorizzazioni per qualsiasi valore per l'impostazione specificata. Ad esempio, se si seleziona il gruppo di livello di accesso Lettura di Amazon EC2, allora le operazioni nella policy supportano il tipo di risorsa `instance`. Devi fornire la regione, l'account e InstanceIdi valori per la tua risorsa. Se si fornisce l'ID account ma si seleziona Any (Qualsiasi) per la regione e l'ID istanza, la policy concede le autorizzazioni a qualsiasi istanza dell'account.
- Digita o incolla l'ARN: puoi possibile specificare le risorse in base ai relativi Amazon Resource Name (ARN). È possibile includere caratteri jolly ***** in qualsiasi campo dell'ARN (tra ogni coppia di due punti). Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Resource.

Autorizzazioni nell'editor visivo

Per impostazione predefinita, la policy creata tramite l'editor visuale permette le operazioni scelte. Per rifiutare invece le operazioni scelte, selezionare Switch to deny permissions (Passa a rifiuto autorizzazioni). Poiché le richieste vengono rifiutate per impostazione predefinita, si consiglia come best practice di sicurezza di permettere le autorizzazioni solo alle operazioni e alle risorse necessarie per un utente. È necessario creare un'istruzione per rifiutare le autorizzazioni solo se si desidera ignorare separatamente un'autorizzazione permessa da un'altra istruzione o policy. Si consiglia di limitare al minimo il numero di autorizzazioni di rifiuto perché possono aumentare la difficoltà di risoluzione dei problemi relative alle autorizzazioni. Per ulteriori informazioni sulla logica di valutazione della policy IAM, consulta Logica di valutazione delle policy.

> **Note**
>
> Per impostazione predefinita, solo Utente root dell'account AWS chi ha accesso a tutte le risorse di quell'account. Pertanto, se non è stato effettuato l'accesso come utente root, è necessario disporre delle autorizzazioni concesse da una policy.

Specifica di più servizi nell'editor visivo

Quando si utilizza l'editor visivo per creare una policy, è possibile selezionare solo un servizio alla volta. Si tratta di una best practice consigliata in quanto l'editor visivo consente in questo modo di selezionare tra le operazioni per tale singolo servizio. Quindi si sceglie tra le risorse supportate da tale servizio e le operazioni selezionate. Ciò semplifica la creazione e la risoluzione dei problemi della policy.

Se si ha familiarità con la sintassi JSON, è anche possibile usare un carattere jolly (*) per specificare manualmente più servizi. Ad esempio, digitare **`Code*`** per fornire le autorizzazioni per tutti i servizi che iniziano con `Code`, ad esempio `CodeBuild` e `CodeCommit`. Tuttavia, è necessario digitare gli ARN della risorsa e le risorse per completare le policy. Inoltre, quando si salva la policy, potrebbe venire ristrutturata per includere ciascun servizio in un blocco di autorizzazioni separate.

In alternativa, per utilizzare una sintassi JSON (ad esempio, i caratteri jolly) per i servizi, crea, modifica e salva la policy utilizzando l'opzione dell'editor JSON.

Riduzione delle dimensioni della policy nell'editor visivo

Quando utilizzi l'editor visivo per creare una policy, IAM crea un documento JSON per archiviare la policy. È possibile visualizzare questo documento passando all'opzione dell'editor JSON. Se questo documento JSON supera il limite di dimensioni di una policy, l'editor visivo visualizza un messaggio di errore e non permette di esaminare e salvare la policy. Per visualizzare i limiti di IAM per le dimensioni di una policy gestita, consulta Limiti di caratteri di IAM e STS.

Per ridurre le dimensioni delle policy nell'editor visivo, modificare la policy o spostare blocchi di autorizzazioni in un'altra policy. Il messaggio di errore include il numero di caratteri che il documento di policy contiene ed è possibile utilizzare queste informazioni per ridurre le dimensioni della policy in modo più semplice.

Correzione di servizi, operazioni o tipi di risorse non riconosciuti nell'editor visivo

Quando si crea o si modifica una policy nell'editor visivo, è possibile che venga visualizzato un avviso che indica che la policy include un servizio, un'operazione o un tipo di risorsa non riconosciuti.

> **Note**
>
> IAM rivede i nomi di servizio, le operazioni e i tipi di risorse per i servizi che supportano i riepiloghi della policy. Tuttavia, il riepilogo della policy può includere un valore di risorse o una condizione che non esiste. Esegui sempre un test delle policy tramite il simulatore di policy.

Se la policy include servizi, operazioni o tipi di risorse non riconosciuti, si è verificato uno dei seguenti errori:

- Servizio di anteprima: i servizi in anteprima non supportano l'editor visivo. Se si partecipa all'anteprima, è possibile ignorare l'avviso e continuare, anche se è necessario digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.
- Servizio personalizzato: i servizi personalizzati non supportano l'editor visivo. Se si utilizza un servizio personalizzato, è possibile ignorare l'avviso e continuare, anche se è necessario digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

- Il servizio non supporta l'editor visivo: se la policy include un servizio con disponibilità a livello generale (GA) non supportato dall'editor visivo, è possibile ignorare l'avviso e continuare, anche se è necessario digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

  I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, significa che il servizio non supporta l'editor visivo. Per informazioni su come richiedere supporto per l'editor visivo o per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- L'operazione non supporta l'editor visivo: se la policy include un servizio supportato con un'operazione non supportata, è possibile ignorare l'avviso e continuare, anche se è necessario digitare manualmente le operazioni e gli ARN delle risorse per completare la policy. In alternativa, per digitare o incollare un documento della policy JSON è possibile scegliere l'opzione dell'editor JSON.

  Se la policy include un servizio supportato con un'operazione non supportata, il servizio non supporta completamente l'editor visivo. Per informazioni su come richiedere supporto per l'editor visivo o per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- Il tipo di risorsa non supporta l'editor visivo: se la policy include un'operazione supportata con un tipo di risorsa non supportato, è possibile ignorare l'avviso e continuare. Tuttavia, IAM non è in grado di confermare di aver incluso risorse per tutte le operazioni selezionate e potrebbero venire visualizzati avvisi aggiuntivi.
- Refuso: quando si digita manualmente un servizio, un'operazione o una risorsa nell'editor visivo, è possibile che venga creata una policy che include un errore di battitura. Per evitare che accada, ti consigliamo di utilizzare l'editor visivo selezionando dall'elenco di servizi e operazioni, quindi completare la sezione delle risorse in base alle richieste. Tuttavia, se un servizio non supporta completamente l'editor visivo, potrebbe essere necessario digitare manualmente le parti della policy.

  Se si è certi che la policy non contenga nessuno degli errori sopra riportati, è possibile che includa un refuso. Cercare nomi di servizi, operazioni e tipi di risorsa errati. Ad esempio, è possibile che si digiti `s2` anziché `s3` o `ListMyBuckets` anziché `ListAllMyBuckets`. Un altro errore di battitura comune è l'inclusione di testo non necessario negli ARN, come `arn:aws:s3: : :*`, oppure la mancanza dei due punti nelle operazioni come `iam.CreateUser`. È possibile valutare una

policy che potrebbe contenere refusi scegliendo Successivo per rivedere il riepilogo della policy e verificare se la policy fornisce le autorizzazioni previste.

Risoluzione dei problemi tramite i riepiloghi delle policy

È possibile individuare e risolvere i problemi relativi ai riepiloghi delle policy.

Riepilogo della policy mancante

La console IAM include tabelle di riepilogo di policy che descrivono il livello di accesso, le risorse e le condizioni concesse o negate per ciascun servizio in una policy. Le policy sono riassunte in tre tabelle: riepilogo della policy, riepilogo del servizio e riepilogo dell'operazione. La tabella riepilogo della policy include un elenco di servizi e riepiloghi delle autorizzazioni definite dalla policy scelta. È possibile visualizzare il riepilogo delle policy per qualsiasi policy associata a un'entità nella pagina Dettagli della policy relativa a tale policy. È possibile visualizzare il riepilogo della policy per le policy gestite nella pagina Policies (Policy). Se non AWS è possibile visualizzare un riepilogo di una policy, viene visualizzato il documento relativo alla policy JSON anziché il riepilogo e viene visualizzato il seguente errore:

Impossibile generare un riepilogo per questa policy. Puoi comunque visualizzare o modificare il documento di policy JSON.

Se la policy non include un riepilogo, si è verificato uno dei seguenti errori:

- Elemento della policy non supportato: IAM non supporta la generazione di riepiloghi di policy per le policy che includono uno dei seguenti elementi di policy:
  - `Principal`
  - `NotPrincipal`
  - `NotResource`
- Nessuna autorizzazione di policy: se una policy non fornisce le autorizzazioni valide, non è possibile generare il riepilogo della policy. Ad esempio, se una policy include una singola istruzione con l'elemento `"NotAction": "*"`, si permette l'accesso a tutte le operazioni ad eccezione di "tutte le operazioni" (*). Questo significa che concede accesso `Deny` o `Allow` a nulla.

> **Note**
>
> Prestare attenzione all'utilizzo di elementi di policy quali `NotPrincipal`, `NotAction` e `NotResource`. Per ulteriori informazioni sull'utilizzo degli elementi delle policy, consultare Documentazione di riferimento degli elementi delle policy JSON IAM.

È possibile creare una policy che non fornisce autorizzazioni valide se si forniscono servizi e risorse non corrispondenti. Ciò può verificarsi se si specificano operazioni in un servizio e risorse di un altro servizio. In questo caso, viene visualizzato il riepilogo della policy. L'unica indicazione che si è verificato un problema è che la colonna delle risorse nel riepilogo può includere una risorsa di un servizio diverso. Se questa colonna include una risorsa non corrispondente, è necessario verificare se ci sono errori nella policy. Per verificare meglio le policy, eseguire sempre un test tramite il simulatore di policy.

Il riepilogo della policy include servizi, operazioni o tipi di risorse non riconosciuti

Nella console IAM, se un riepilogo di policyinclude un simbolo di avvertenza
( ),
la policy potrebbe includere un tipo di servizio, azione o risorsa non riconosciuto. Per ulteriori informazioni sulle avvertenze in un riepilogo della policy, consultare Riepilogo della policy (elenco di servizi).

> **Note**
>
> IAM rivede i nomi di servizio, le operazioni e i tipi di risorse per i servizi che supportano i riepiloghi della policy. Tuttavia, il riepilogo della policy può includere un valore di risorse o una condizione che non esiste. Esegui sempre un test delle policy tramite il simulatore di policy.

Se la policy include servizi, operazioni o tipi di risorse non riconosciuti, si è verificato uno dei seguenti errori:

- Servizio di anteprima: i servizi in anteprima non supportano i riepiloghi di policy.
- Servizio personalizzato: i servizi personalizzati non supportano i riepiloghi di policy.

- Il servizio non supporta riepiloghi: se la policy include un servizio disponibile a livello generale (GA) che non supporta riepiloghi di policy, allora il servizio viene incluso nella sezione Servizi non riconosciuti della tabella di riepilogo della policy. I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, allora significa che il servizio non supporta i riepiloghi di policy IAM. Per informazioni su come richiedere supporto per il riepilogo della policy per un servizio con disponibilità generale, consultare Il servizio non supporta i riepiloghi delle policy IAM.
- L'operazione non supporta i riepiloghi: se la policy include un servizio supportato con un'operazione non supportata, allora l'operazione viene inclusa nella sezione Operazioni non riconosciute della tabella di riepilogo del servizio. Per ulteriori informazioni sulle avvertenze in un riepilogo di servizio, consultare Riepilogo del servizio (elenco di operazioni).
- Il tipo di risorsa non supporta i riepiloghi: se la policy include un'operazione supportata con un tipo di risorsa non supportato, allora la risorsa viene inclusa nella sezione Tipi di risorse non riconosciuti della tabella di riepilogo del servizio. Per ulteriori informazioni sulle avvertenze in un riepilogo di servizio, consultare Riepilogo del servizio (elenco di operazioni).
- Errore di battitura: AWS verifica che il codice JSON sia sintatticamente corretto e che la policy non includa errori di battitura o altri errori come parte della convalida della politica.

> **Note**
>
> Come best practice, ti consigliamo di utilizzare IAM Access Analyzer per convalidare le tue policy IAM e garantire autorizzazioni sicure e funzionali. Consigliamo di aprire le policy esistenti e rivedere e risolvere eventuali suggerimenti di convalida della policy.

Il servizio non supporta i riepiloghi delle policy IAM

Quando un'operazione o un servizio disponibile a livello generale (GA) non è riconosciuto dai riepiloghi di policy IAM o dall'editor visivo, è possibile che il servizio non supporti queste caratteristiche. I servizi disponibili a livello generale sono servizi che vengono rilasciati per il pubblico e che non sono servizi di anteprima o personalizzati. Se un servizio non riconosciuto è disponibile a livello generale e il nome è scritto correttamente, significa che il servizio non supporta queste caratteristiche. Se la policy include un servizio supportato con un'operazione non supportata, il servizio non supporta completamente il riepilogo della policy IAM.

Come richiedere che un servizio aggiunga il supporto per l'editor visivo o per il riepilogo della policy IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Individuare la policy che include il servizio non supportati:
   - Se la policy è una policy gestita, selezionare Policies (Policy) nel riquadro di navigazione. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
   - Se la policy è una policy inline collegata all'utente, selezionare Users (Utenti) nel riquadro di navigazione. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare. Nella tabella di policy per l'utente, espandere l'intestazione per il riepilogo della policy che si desidera visualizzare.
3. Nella parte sinistra del AWS Management Console piè di pagina, scegli Feedback. Nella casella Feedback per IAM, digita **`I request that the <ServiceName> service add support for IAM policy summaries and the visual editor`**. Se si desidera che più di un servizio supporti i riepiloghi, digitare **`I request that the <ServiceName1>, <ServiceName2>, and <ServiceName3> services add support for IAM policy summaries and the visual editor`**.

Come richiedere che un servizio aggiunga il supporto per l'editor visivo o per il riepilogo della policy IAM per un'operazione mancante

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Individuare la policy che include il servizio non supportati:
   - Se la policy è una policy gestita, selezionare Policies (Policy) nel riquadro di navigazione. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
   - Se la policy è una policy inline collegata all'utente, selezionare Users (Utenti) nel riquadro di navigazione. Nell'elenco di utenti, selezionare il nome dell'utente la cui policy si desidera visualizzare. Nella tabella delle policy per l'utente, selezionare il nome della policy che cui si desidera visualizzare per espandere il riepilogo della policy.
3. Nel riepilogo della policy, selezionare il nome del servizio che include un'operazione non supportata.

4. Nella parte sinistra del AWS Management Console piè di pagina, scegli Feedback. Nella casella Feedback per IAM, digita **`I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName> action`**. Se si desidera segnalare più di un'operazione non supportata, digitare **`I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName1>, <ActionName2>, and <ActionName3> actions`**.

Per richiedere che un altro servizio includa operazioni mancanti, ripetere gli ultimi tre passaggi.

La policy non concede le autorizzazioni previste

Per assegnare le autorizzazioni a un utente, gruppo, ruolo o risorsa, è necessario creare una policy, ovvero un documento che definisca le autorizzazioni. Il documento della policy include i seguenti elementi:

- Effetto: indica se la policy consente o nega l'accesso
- Azione: l'elenco delle operazioni consentite o rifiutate dalla policy
- Risorsa: l'elenco delle risorse in cui possono essere eseguite le operazioni
- Condizione (facoltativo): le circostanze in base alle quali la policy concede l'autorizzazione

Per informazioni su questi elementi di policy, consultare Documentazione di riferimento degli elementi delle policy JSON IAM.

Per garantire l'accesso, la policy deve definire un'operazione con una risorsa supportata. Se la policy include anche una condizione, tale condizione deve includere una chiave di condizione globale o deve essere applicata all'operazione. Per scoprire quali risorse sono supportate da un'operazione, consulta la documentazione di AWS relativa al servizio. Per sapere quali condizioni sono supportate da un'azione, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.

Per scoprire se la policy definisce un'operazione, una risorsa o una condizione che non concede autorizzazioni, puoi visualizzare il riepilogo della policy per la policy utilizzando la console IAM all'indirizzo https://console.aws.amazon.com/iam/. È possibile utilizzare riepiloghi di policy per identificare e risolvere i problemi della policy.

Esistono diversi motivi per cui un elemento potrebbe non concedere autorizzazioni nonostante sia definito nella policy IAM:

- Un'operazione è definita senza una risorsa applicabile
- Una risorsa è definita senza un'operazione applicabile
- Una condizione è definita senza un'operazione applicabile

Per visualizzare esempi di riepiloghi di policy che includono avvisi, consultare the section called "Riepilogo della policy (elenco di servizi)".

Un'operazione è definita senza una risorsa applicabile

La policy di seguito definisce tutte le operazioni `ec2:Describe*` e definisce una risorsa specifica. Nessuna delle operazioni `ec2:Describe` viene permessa perché nessuna di tali operazioni supporta le autorizzazioni a livello di risorsa. Le autorizzazioni a livello di risorsa indicano che l'operazione supporta le risorse che utilizzano gli ARN nell'elemento `Resource` della policy. Se un'operazione non supporta le autorizzazioni a livello di risorsa, tale istruzione della policy deve utilizzare un carattere jolly (`*`) nell'elemento `Resource`. Per scoprire i servizi che supportano le autorizzazioni a livello di risorsa, consultare AWS servizi che funzionano con IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "arn:aws:ec2:us-east-2:ACCOUNT-ID:instance/*"
    }]
}
```

Questa policy non fornisce alcuna autorizzazione e il riepilogo della policy include il seguente errore:

```
This policy does not grant any permissions. To grant access, policies must
have an action that has an applicable resource or condition.
```

Per correggere la policy, è necessario utilizzare `*` nell'elemento `Resource`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
    }]
```

```
}
```

Una risorsa è definita senza un'operazione applicabile

La policy di seguito definisce una risorsa del bucket Amazon S3 ma non include un'operazione S3 che può essere eseguita su tale risorsa. Questa politica garantisce inoltre l'accesso completo a tutte le CloudFront azioni di Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "cloudfront:*",
        "Resource": [
            "arn:aws:cloudfront:*",
            "arn:aws:s3:::examplebucket"
        ]
    }]
}
```

Questa politica fornisce le autorizzazioni per tutte le CloudFront azioni. Tuttavia, poiché la policy definisce le risorse `examplebucket` di S3 senza definire alcuna operazione S3, il riepilogo della policy include il seguente avviso:

```
This policy defines some actions, resources, or conditions that do not
provide permissions. To grant access, policies must have an action that has
an applicable resource or condition.
```

Per correggere questa policy per fornire autorizzazioni del bucket S3, è necessario definire operazioni S3 che possono essere eseguite su una risorsa bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "cloudfront:*",
            "s3:CreateBucket",
            "s3:ListBucket*",
            "s3:PutBucket*",
            "s3:GetBucket*"
```

```
        ],
        "Resource": [
            "arn:aws:cloudfront:*",
            "arn:aws:s3:::examplebucket"
        ]
    }]
}
```

In alternativa, per correggere questa politica e fornire solo le CloudFront autorizzazioni, rimuovi la risorsa S3.

Una condizione è definita senza un'operazione applicabile

La policy di seguito definisce due operazioni Amazon S3 per tutte le risorse S3, se il prefisso S3 è uguale a `custom` e l'ID della versione è uguale a `1234`. Tuttavia, la chiave della condizione `s3:VersionId` viene utilizzata per il tagging della versione dell'oggetto e non è supportata dalle operazioni bucket definite. Per sapere quali condizioni sono supportate da un'azione, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e segui il link alla documentazione del servizio per le chiavi di condizione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
```

```
}
```

Questa policy fornisce le autorizzazioni per l'operazione `s3:ListBucketVersions` e l'operazione `s3:ListBucket` se il nome del bucket include il prefisso `custom`. Tuttavia, poiché la condizione `s3:VersionId` non è supportata da nessuna delle operazioni definite, il riepilogo della policy include il seguente errore:

`This policy does not grant any permissions. To grant access, policies must have an action that has an applicable resource or condition.`

Per correggere questa policy e utilizzare il tagging della versione dell'oggetto S3, è necessario definire un'operazione S3 che supporti la chiave della condizione `s3:VersionId`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:GetObjectVersion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
}
```

Questa policy fornisce le autorizzazioni per ogni operazione e condizione nella policy. Tuttavia, la policy non fornisce nessuna autorizzazione perché non esiste un caso in cui una singola operazione corrisponda a entrambe le condizioni. Al contrario, è necessario creare due dichiarazioni separate che includano ciascuna solo le operazioni con le condizioni a cui si applicano.

Per correggere questa policy, è necessario creare due istruzioni. La prima istruzione include le operazioni che supportano la condizione `s3:prefix` e la seconda istruzione include le operazioni che supportano la condizione `s3:VersionId`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "custom"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObjectVersion",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:VersionId": "1234"
                }
            }
        }
    ]
}
```

Risoluzione dei problemi di gestione delle policy

È possibile individuare e risolvere i problemi correlati alla gestione delle policy.

Collegamento o scollegamento di una policy in un account IAM

Alcune politiche AWS gestite sono collegate a un servizio. Queste policy vengono utilizzate solo con un ruolo collegato a un servizio per tale servizio. Nella console IAM, quando si visualizza la pagina Dettagli della policy per una policy, sarà presente un banner per indicare che la policy è

collegata a un servizio. Non è possibile collegare questa policy a un utente, un gruppo o un ruolo all'interno di IAM. Quando si crea un ruolo collegato al servizio per il servizio, questa policy viene automaticamente collegata al nuovo ruolo. Poiché la policy è obbligatoria, non è possibile distaccare la policy dal ruolo collegato al servizio.

Modifica delle policy per le identità IAM in base alla loro attività

È possibile aggiornare le policy per le identità IAM (utenti, gruppi e ruoli) in base alla loro attività. A tale scopo, visualizza gli eventi del tuo account nella CloudTrail Cronologia degli eventi. CloudTrail i registri degli eventi includono informazioni dettagliate sugli eventi che puoi utilizzare per modificare le autorizzazioni della politica. È possibile che un utente o un ruolo stia tentando di eseguire un'azione in AWS e tale richiesta venga rifiutata. In tal caso, puoi valutare se l'utente o il ruolo deve disporre dell'autorizzazione per eseguire l'operazione. In questo caso, è possibile aggiungere alla policy l'operazione e anche l'ARN della risorsa a cui cerca di accedere. In alternativa, se l'utente o il ruolo dispone di autorizzazioni che non utilizza, è possibile eliminare tali autorizzazioni dalla policy. Verifica che le policy concedano i privilegi minimi necessari per eseguire solo le operazioni necessarie. Per ulteriori informazioni sull'utilizzo CloudTrail, consulta Visualizzazione CloudTrail degli eventi nella CloudTrail console nella Guida per l'AWS CloudTrail utente.

Risoluzione dei problemi relativi ai documenti di policy JSON

È possibile individuare e risolvere i problemi relativi ai documenti di policy JSON.

Convalida delle policy

Quando crei o si modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni sui controlli di policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Non ho autorizzazioni per la convalida di policy nell'editor JSON

In AWS Management Console, potresti ricevere il seguente errore se non disponi delle autorizzazioni per visualizzare i risultati di convalida delle policy di IAM Access Analyzer:

```
You need permissions. You do not have the permissions required to perform
this operation. Ask your administrator to add permissions.
```

Per risolvere questo errore, chiedere all'amministratore di aggiungere l'autorizzazione `access-analyzer:ValidatePolicy` per proprio conto.

Più di un oggetto di policy JSON

Una policy IAM deve essere costituita da un solo oggetto JSON. È possibile denotare un oggetto racchiudendolo tra parentesi graffe { }. Sebbene sia possibile nidificare altri oggetti all'interno di un oggetto JSON incorporando ulteriori parentesi graffe { } all'interno della coppia esterna, una policy può contenere solo una coppia più esterna di parentesi graffe { }. L'esempio seguente non è corretto perché contiene due oggetti al livello superiore (evidenziati in *rosso*):

```
{
      "Version": "2012-10-17",
      "Statement":
      {
         "Effect":"Allow",
         "Action":"ec2:Describe*",
         "Resource":"*"
      }
   }
   {
     "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket/*"
     }
   }
```

Tuttavia, è possibile soddisfare l'intenzione dell'esempio precedente con l'uso di una corretta grammatica di policy. Anziché includere due oggetti di policy completi ciascuno con il proprio elemento `Statement`, è possibile combinare i due blocchi in un singolo elemento `Statement`. L'elemento `Statement` dispone di una serie di due oggetti come valore, come mostrato nell'esempio seguente (evidenziato in grassetto):

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "ec2:Describe*",
          "Resource":" *"
```

```
        },
        {
          "Effect": "Allow",
          "Action": "s3:*",
          "Resource": "arn:aws:s3:::my-bucket/*"
        }
      ]
    }
```

Più di un elemento di istruzione JSON

Questo errore potrebbe apparentemente sembrare una variazione della sezione precedente. Tuttavia, sintatticamente si tratta di un altro tipo di errore. L'esempio seguente include un solo oggetto della policy come indicato da una singola coppia di parentesi graffe { } al livello più alto. Tuttavia, quell'oggetto contiene due elementi `Statement` al suo interno.

Una policy IAM deve contenere solo un elemento `Statement`, che include il nome (`Statement`) che appare alla sinistra di due punti, seguito dal rispettivo valore sulla destra. Il valore di un elemento `Statement` deve essere un oggetto, contrassegnato da parentesi graffe {}, che contiene un elemento `Effect`, un elemento `Action` e un elemento `Resource`. L'esempio seguente non è corretto perché contiene due elementi `Statement` nell'oggetto policy (evidenziato in *rosso*):

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
      },
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket/*"
      }
    }
```

Un oggetto valore può essere una matrice di oggetti a valore multiplo. Per risolvere questo problema, combinare i due elementi `Statement` in un unico elemento con una matrice di oggetti, come mostrato nell'esempio seguente (evidenziato in grassetto):

```
{
```

```
      "Version": "2012-10-17",
      "Statement": [    {
          "Effect": "Allow",
          "Action": "ec2:Describe*",
          "Resource":"*"
        },
        {
          "Effect": "Allow",
          "Action": "s3:*",
          "Resource": "arn:aws:s3:::my-bucket/*"
        }
     ]
    }
```

Il valore dell'elemento `Statement` è una matrice di oggetti. La matrice in questo esempio è costituita da due oggetti, ognuno dei quali è di per sé un valore corretto per un elemento `Statement`. Ogni oggetto nella matrice è separato da virgole.

Più di un elemento Effect, Action o Resource in un elemento di istruzione JSON

Sul lato valore della coppia nome/valore `Statement`, l'oggetto deve essere composto da un solo elemento `Effect`, un elemento `Action` e un elemento `Resource`. La policy seguente non è corretta poiché include due elementi `Effect` nel valore dell'oggetto `Statement`:

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Deny",
        "Effect": "Allow",
        "Action": "ec2:* ",
        "Resource": "*"
      }
    }
```

> **Note**
>
> Il motore di policy non permette tali errori nelle policy nuove o modificate. Tuttavia, il motore di policy permettere le policy che sono state salvate prima che il motore fosse aggiornato. Il comportamento delle policy esistenti con l'errore è il seguente:

- Più elementi `Effect`: viene osservato solo l'ultimo elemento `Effect`. Gli altri valori vengono ignorati.
- Più elementi `Action`: tutti gli elementi `Action` vengono combinati internamente e trattati come se fossero un unico elenco.
- Più elementi `Resource`: tutti gli elementi `Resource` vengono combinati internamente e trattati come se fossero un unico elenco.

Il motore di policy non permette di salvare alcuna policy con errori di sintassi. È necessario correggere gli errori nella policy prima di poterla salvare. Consigliamo di rivedere e correggere tutti i suggerimenti di convalida delle policy per le policy.

In ogni caso, la soluzione consiste nel rimuovere l'elemento aggiuntivo. Per elementi `Effect`, è semplice: se si desidera che l'esempio precedente rifiuti le autorizzazioni per le istanze Amazon EC2, è necessario rimuovere la riga `"Effect": "Allow",` dalla policy come segue:

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Deny",
        "Action": "ec2:* ",
        "Resource": "*"
      }
    }
```

Tuttavia, se l'elemento duplicato è `Action` oppure `Resource`, la risoluzione può essere più complicata. Potrebbero essere presenti più operazioni per cui si desidera permettere (o rifiutare) l'autorizzazione oppure si potrebbe voler controllare l'accesso a più risorse. Ad esempio, l'esempio seguente non è corretto perché sono presenti più elementi `Resource` (evidenziati in *rosso*):

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket",
        "Resource": "arn:aws:s3:::my-bucket/*"
      }
```

```
    }
```

Tutti gli elementi necessari in un oggetto valore di un elemento `Statement` può essere presente una sola volta. La soluzione consiste nel posizionare ogni valore in una matrice. L'esempio che segue illustra questa operazione separando i due elementi risorsa in un elemento `Resource` con una matrice come valore oggetto (evidenziata in grassetto):

```
{
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*"
        ]
      }
    }
```

Elemento versione JSON mancante

Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called “Controllo delle versioni delle policy IAM”.

Man mano che AWS le funzionalità si evolvono, vengono aggiunte nuove funzionalità alle policy IAM per supportare tali funzionalità. A volte, un aggiornamento della sintassi della policy include un nuovo numero di versione. Se si utilizzano le caratteristiche più recenti della grammatica di policy nella policy, è necessario indicare al motore di analisi delle policy quale versione si sta utilizzando. La versione di policy predefinita è "2008-10-17". Se si desidera utilizzare qualsiasi funzione di policy introdotta successivamente, è necessario specificare il numero di versione che supporta la funzionalità desiderata. Si consiglia di includere sempre il numero di versione della sintassi di policy più recente, che è al momento `"Version": "2012-10-17"`. Ad esempio, la policy seguente non è corretta perché utilizza una variabile di policy `${...}` nell'ARN per una risorsa. Tuttavia, se non è in

grado di specificare una versione della sintassi di policy che supporta variabili di policy (evidenziate in *rosso*):

```
{
  "Statement":
  {
    "Action": "iam:*AccessKey*",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::123456789012:user/${aws:username}"
  }
}
```

Aggiungendo un elemento `Version` nella parte superiore della policy con il valore `2012-10-17`, la prima versione di API IAM che supporta variabili di policy, è possibile risolvere il problema (evidenziato in grassetto):

```
{
  "Version": "2012-10-17",
  "Statement":
  {
    "Action": "iam:*AccessKey*",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::123456789012:user/${aws:username}"
  }
}
```

Risoluzione dei problemi relativi alle chiavi di sicurezza FIDO

Utilizza le informazioni contenute in questa pagina per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo delle chiavi di sicurezza FIDO2.

Argomenti

Non riesco ad abilitare la chiave di sicurezza FIDO

Consulta le seguenti soluzioni a seconda del tuo stato come amministratore di sistema o utente IAM.

Utenti IAM

Se non riesci ad abilitare la chiave di sicurezza FIDO, verifica quanto segue:

- Stai utilizzando una configurazione supportata?

  Per informazioni sui dispositivi e sui browser che è possibile utilizzare con WebAuthn e AWS, vedereConfigurazioni supportate per l'utilizzo di chiavi di accesso e chiavi di sicurezza.

- Stai utilizzando Mozilla Firefox?

  Le versioni correnti di Firefox sono WebAuthn supportate per impostazione predefinita. Per abilitare il supporto per WebAuthn Firefox, procedi come segue:

  1. Nella barra degli indirizzi di Firefox, digitare **`about:config`**.
  2. Nel barra di ricerca della schermata visualizzata, digitare **`webauthn`**.
  3. Scegli security.webauth.webauthn e modificane il valore su true.

- Stai utilizzando plug-in di browser?

  AWS non supporta l'uso di plugin per aggiungere il supporto al WebAuthn browser. Utilizza invece un browser che offra il supporto nativo dello WebAuthn standard.

  Anche se utilizzi un browser supportato, potresti avere un plug-in con WebAuthn cui è incompatibile. Un plug-in non compatibile potrebbe impedirti di abilitare e utilizzare la chiave di sicurezza conforme a FIDO. È consigliabile disabilitare qualsiasi plug-in che può risultare incompatibile e riavviare il browser. Quindi riprova ad abilitare la chiave di sicurezza FIDO.

- Disponi delle autorizzazioni appropriate?

  Se non hai nessuno dei problemi descritti precedentemente, è possibile che non disponi delle autorizzazioni appropriate. Contatta l'amministratore di sistema.

Amministratori di sistema

Se sei un amministratore e i tuoi utenti IAM non sono in grado di abilitare le proprie chiavi di sicurezza FIDO nonostante utilizzino una configurazione supportata, assicurati che dispongano delle

autorizzazioni appropriate. Per un esempio dettagliato, consulta Tutorial IAM: consentire agli utenti di gestire le proprie credenziali e impostazioni MFA.

Non riesco a effettuare l'accesso utilizzando la chiave di sicurezza FIDO

Se sei un utente IAM e non riesci ad accedere AWS Management Console utilizzando la tua chiave di sicurezza FIDO, consulta innanzitutto. Configurazioni supportate per l'utilizzo di chiavi di accesso e chiavi di sicurezza Se utilizzi una configurazione supportata, ma non puoi effettuare l'accesso, contatta l'amministratore di sistema per ricevere assistenza.

Ho perso o danneggiato la mia chiave di sicurezza FIDO

Puoi assegnare a un utente fino a otto dispositivi MFA in qualsiasi combinazione dei tipi di MFA attualmente supportati. Con più dispositivi MFA, per accedere alla AWS Management Consoleè necessario un solo dispositivo MFA. La sostituzione di una chiave di sicurezza FIDO è analoga alla sostituzione di un token TOTP hardware. Per informazioni su cosa fare se si perde o si rompe qualsiasi tipo di dispositivo MFA, consulta Cosa fare se un dispositivo MFA viene smarrito o smette di funzionare?.

Altri problemi.

Se hai un problema con le chiavi di sicurezza FIDO non descritto qui, procedi in uno dei seguenti modi:

- Utenti IAM: contattare l'amministratore di sistema.
- Utenti root Account AWS : contattare AWS Support.

Risoluzione dei problemi dei ruoli IAM

Utilizza le informazioni contenute in questa pagina per eseguire la diagnosi e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di ruoli IAM.

Argomenti

- Non sono autorizzato a eseguire: iam: PassRole
- Perché non posso assumere un ruolo con una sessione di 12 ore? (AWS CLI, AWS API)
- Viene visualizzato un errore quando provo a passare da un ruolo a un altro nella console IAM
- Il mio ruolo ha un policy che mi consente di eseguire un'operazione, ma ricevo "Accesso negato"
- Il servizio non ha creato la versione delle policy predefinite del ruolo
- Non esiste un caso d'uso per un ruolo di servizio nella console

Non è possibile assumere un ruolo

Verifica quanto segue:

- Per consentire agli utenti di assumere nuovamente il ruolo corrente all'interno di una sessione di ruolo, specificare il ruolo ARN o Account AWS ARN come principale nella politica di attendibilità dei ruoli. Servizi AWS che forniscono risorse di elaborazione come Amazon EC2, Amazon ECS, Amazon EKS e Lambda forniscono credenziali temporanee e aggiornano automaticamente tali credenziali. Ciò garantisce di disporre sempre di un set di credenziali valido. Per questi servizi, non è necessario riassumere il ruolo attuale per ottenere credenziali temporanee. Tuttavia, se intendi passare tag di sessione o una Policy di sessione, devi riassumere il ruolo attuale. Per informazioni su come modificare una politica di attendibilità dei ruoli per aggiungere il ruolo principale ARN o Account AWS ARN, vedere. Modifica di una policy di attendibilità del ruolo (Console)
- Quando assumi un ruolo utilizzando il AWS Management Console, assicurati di utilizzare il nome esatto del ruolo. I nomi dei ruoli fanno infatti distinzione tra maiuscole e minuscole.
- Quando assumi un ruolo utilizzando l' AWS STS API oppure AWS CLI, assicurati di utilizzare il nome esatto del tuo ruolo nell'ARN. I nomi dei ruoli fanno infatti distinzione tra maiuscole e minuscole.
- Verifica che la policy IAM conceda l'autorizzazione per chiamare `sts:AssumeRole` per il ruolo che desideri assumere. L'elemento `Action` della policy IAM deve consentire di chiamare l'operazione `AssumeRole`. Inoltre, l'elemento `Resource` della policy IAM deve specificare il ruolo che desideri assumere. Ad esempio, l'elemento `Resource` può specificare un ruolo in base all'Amazon Resource Name (ARN) o utilizzando un carattere jolly (*). Ad esempio, almeno una policy applicabile è necessaria per concedere le autorizzazioni simili a quanto segue:

```
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
```

- Verifica che l'identità IAM sia taggata con eventuali tag richiesti dalla policy IAM. Ad esempio, nella seguente policy di autorizzazione, l'elemento `Condition` richiede che il principale richiedente l'assunzione del ruolo debba avere un determinato tag. Al principale deve essere applicato il tag `department = HR` o `department = CS`. In caso contrario, non può assumere quel ruolo. Per ulteriori informazioni sul tagging di utenti e ruoli IAM, consulta the section called “Tagging delle risorse IAM”.

```
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": "*",
"Condition": {"StringEquals": {"aws:PrincipalTag/department": [
        "HR",
        "CS"
    ]}}
```

- Verificare di soddisfare tutte le condizioni specificate nella policy di affidabilità del ruolo. Una `Condition` può specificare una data di scadenza, un ID esterno o che una richiesta deve provenire solo da indirizzi IP specifici. Considera l'esempio seguente: se la data corrente è qualsiasi momento dopo la data specifica, la policy non corrisponde mai e non è in grado di concedere l'autorizzazione per assumere il ruolo.

```
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
"Condition": {
    "DateLessThan" : {
        "aws:CurrentTime" : "2016-05-01T12:00:00Z"
    }
}
```

- Verifica che l'entità Account AWS da cui stai chiamando `AssumeRole` sia un'entità attendibile per il ruolo che stai assumendo. Le entità affidabili vengono definite come `Principal` in una policy di affidabilità del ruolo. L'esempio seguente è una policy di affidabilità collegata al ruolo che desideri assumere. In questo esempio, l'ID account con l'utente IAM con il quale hai effettuato l'accesso deve essere 123456789012. Se il numero di account non è elencato nell'elemento `Principal` della policy di affidabilità del ruolo, non puoi assumere il ruolo. Ciò è valido indipendentemente dalle autorizzazioni concesse nelle policy di accesso. Notare che la policy di esempio limita le autorizzazioni a operazioni che si verificano tra il 1° luglio 2017 e il 31 dicembre 2017 (UTC), inclusi. Se si effettua l'accesso prima o dopo tali date, la policy non corrisponde e non è possibile assumere il ruolo.

```
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
    "Action": "sts:AssumeRole",
    "Condition": {
      "DateGreaterThan": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
      "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
    }
```

- Identità di origine: gli amministratori possono configurare i ruoli in modo da richiedere alle identità di passare una stringa personalizzata che identifica la persona o l'applicazione in cui esegue le azioni AWS, denominata identità di origine. Verifica se il ruolo assunto richiede l'impostazione di un'identità di origine. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Un nuovo ruolo appare nell'account AWS

Alcuni AWS servizi richiedono l'utilizzo di un tipo di ruolo di servizio univoco collegato direttamente al servizio. Questo ruolo collegato ai servizi è predefinito dal servizio e include tutte le autorizzazioni che il servizio richiede. Ciò rende più semplice la configurazione di un servizio perché non si devono aggiungere manualmente le autorizzazioni necessarie. Per informazioni generali sui ruoli collegati al servizio, consultare Uso di ruoli collegati ai servizi.

Un servizio potrebbe essere già in utilizzo quando inizia a supportare i ruoli collegati al servizio. In questo caso, è possibile ricevere un'e-mail con informazioni su un nuovo ruolo nell'account. Questo ruolo include tutte le autorizzazioni delle quali il servizio ha bisogno per eseguire operazioni a proprio nome. Non bisogna eseguire alcuna operazione per supportare questo ruolo. Tuttavia, non bisogna eliminare il ruolo dall'account. Altrimenti si potrebbero rimuovere le autorizzazioni delle quali il servizio ha bisogno per accedere alle risorse AWS . È possibile visualizzare i ruoli collegati ai servizi nell'account passando la pagina Ruoli IAM della console IAM. Per i ruoli collegati al servizio viene visualizzata l'indicazione (Service-linked role) (Ruolo collegato al servizio) nella colonna Trusted entities (Entità attendibili) della tabella.

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta AWS servizi che funzionano con IAM e cercare i servizi che hanno Sì nella colonna Ruolo collegato ai servizi. Per ulteriori informazioni sull'utilizzo di un ruolo collegato ai servizi per un servizio, selezionare il collegamento Yes (Sì).

Non è possibile modificare o eliminare un ruolo nell' Account AWS

Non è possibile eliminare o modificare le autorizzazioni per un ruolo collegato ai servizi in IAM. Questi ruoli includono trust e autorizzazioni predefiniti richiesti dal servizio per eseguire operazioni a proprio nome. Puoi utilizzare la console o l'API IAM per modificare solo la descrizione di un ruolo collegato al servizio. AWS CLI Puoi visualizzare i ruoli collegati ai servizi nell'account visitando la pagina Ruoli IAM nella console. Per i ruoli collegati al servizio viene visualizzata l'indicazione (Service-linked role) (Ruolo collegato al servizio) nella colonna Trusted entities (Entità attendibili) della tabella. Un banner nella pagina Summary (Riepilogo) del ruolo indica anche che un ruolo è un ruolo collegato ai servizi. È possibile gestire ed eliminare questi ruoli solo attraverso il servizio collegato, se quel servizio supporta l'operazione. Fare attenzione quando si modifica o elimina un ruolo collegato ai servizi poiché tale operazione può rimuovere le autorizzazioni delle quali il servizio ha bisogno per accedere alle risorse AWS .

Per informazioni su quali servizi supportano i ruoli collegati ai servizi, consulta AWS servizi che funzionano con IAM e cercare i servizi che hanno Sì nella colonna Ruolo collegato ai servizi.

Non sono autorizzato a eseguire: iam: PassRole

Quando si crea un ruolo collegato ai servizi, è necessario disporre delle autorizzazioni per inoltrare quel ruolo al servizio. Alcuni servizi creano automaticamente un ruolo collegato ai servizi nell'account quando si esegue un'azione in quel servizio. Ad esempio, Amazon EC2 Auto Scaling crea il ruolo collegato ai servizi `AWSServiceRoleForAutoScaling` la prima volta che si crea un gruppo Auto Scaling. Se si cerca di creare un gruppo Auto Scaling senza l'autorizzazione `PassRole`, si riceve il seguente messaggio di errore:

```
ClientError: An error occurred (AccessDenied) when calling the
PutLifecycleHook operation: User: arn:aws:sts::111122223333:assumed-role/
Testrole/Diego is not authorized to perform: iam:PassRole on resource:
arn:aws:iam::111122223333:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling
```

Per risolvere questo errore, chiedere all'amministratore di aggiungere l'autorizzazione `iam:PassRole` per proprio conto.

Per scoprire i servizi che supportano i ruoli collegati ai servizi, consulta AWS servizi che funzionano con IAM. Per scoprire se un servizio crea automaticamente un ruolo collegato ai servizi, selezionare il link Yes (Sì) per visualizzare la documentazione del ruolo collegato ai servizi per quel servizio.

Perché non posso assumere un ruolo con una sessione di 12 ore? (AWS CLI, AWS API)

Quando si utilizzano le operazioni AWS STS `AssumeRole*` API o `assume-role*` CLI per assumere un ruolo, è possibile specificare un valore per il `DurationSeconds` parametro. Puoi specificare un valore da 900 secondi (15 minuti) fino alla durata massima impostata per la sessione per il ruolo. Se si specifica un valore superiore a questa impostazione, l'operazione ha esito negativo. Questa impostazione può avere un valore massimo di 12 ore. Ad esempio, se si specifica una durata di sessione di 12 ore, ma l'amministratore ha impostato la durata massima di sessione a 6 ore, l'operazione ha esito negativo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.

Se si utilizza l'concatenamento dei ruoli (utilizzando un ruolo per assumere un secondo ruolo), la sessione è limitata a un massimo di un'ora. Se successivamente utilizzi il parametro `DurationSeconds` per fornire un valore superiore a un'ora, l'operazione ha esito negativo.

Viene visualizzato un errore quando provo a passare da un ruolo a un altro nella console IAM

Le informazioni immesse nella pagina Cambia ruolo devono corrispondere a quelle relative al ruolo. In caso contrario, l'operazione non riesce e viene visualizzato il seguente errore:

```
Invalid information in one or more fields. Check your information or
contact your administrator.
```

Se viene visualizzato questo errore, verificare che le seguenti informazioni siano corrette:

- ID account o alias: l' Account AWS ID è un numero di 12 cifre. Il tuo account potrebbe avere un alias, che è un identificativo semplice come il nome della tua azienda, che può essere utilizzato al posto del tuo ID. Account AWS In questo campo è possibile utilizzare l'ID account o l'alias.
- Nome ruolo: i nomi dei ruoli fanno distinzione tra maiuscole e minuscole. L'ID account e il nome del ruolo devono corrispondere a quelli configurati per il ruolo.

Se si continua a ricevere un messaggio di errore, contattare l'amministratore per verificare le informazioni precedenti. La policy di attendibilità del ruolo o la policy dell'utente IAM potrebbe limitare l'accesso. L'amministratore può verificare le autorizzazioni per questi criteri.

Il mio ruolo ha un policy che mi consente di eseguire un'operazione, ma ricevo "Accesso negato"

La sessione del ruolo potrebbe essere limitata dalle policy di sessione. Quando richiedi credenziali di sicurezza temporanee a livello di codice AWS STS, puoi facoltativamente passare policy di sessione in linea o gestite. Le policy di sessione sono policy avanzate che vengono passate come parametro durante la creazione di una sessione temporanea per un ruolo a livello di programmazione. Puoi passare un singolo documento della policy di sessione inline JSON utilizzando il parametro `Policy`. Puoi utilizzare il parametro `PolicyArns` per specificare fino a 10 policy di sessione gestite. Le autorizzazioni della sessione risultanti sono l'intersezione tra le policy basate sull'identità del ruolo e le policy di sessione. In alternativa, se l'amministratore o un programma personalizzato fornisce le credenziali temporanee, potrebbero includere policy di sessione per limitare l'accesso.

Il servizio non ha creato la versione delle policy predefinite del ruolo

Un ruolo di servizio è un ruolo che un servizio assume per eseguire operazioni nel tuo account a tuo nome. Quando si configurano alcuni ambienti di AWS servizio, è necessario definire un ruolo da assumere per il servizio. In alcuni casi, il servizio crea il ruolo di servizio e le relative policy in IAM per tuo conto. Sebbene sia possibile modificare o eliminare il ruolo di servizio e la relativa policy dall'interno di IAM, AWS consiglia di non seguire questa opzione. Il ruolo e il criterio sono destinati solo a tale servizio. Se si modifica il criterio e si imposta un altro ambiente, quando il servizio tenta di utilizzare lo stesso ruolo e criterio, l'operazione potrebbe non riuscire.

Ad esempio, quando si utilizza AWS CodeBuild per la prima volta, il servizio crea un ruolo denominato`codebuild-RWBCore-service-role`. Tale ruolo di servizio utilizza il criterio denominato `codebuild-RWBCore-managed-policy`. Se si modifica il criterio, viene creata una nuova versione che viene salvata come versione predefinita. Se si esegue un'operazione successiva in AWS CodeBuild, il servizio potrebbe tentare di aggiornare la politica. In tal caso, viene visualizzato il seguente errore:

```
codebuild.amazon.com did not create the default version (V2) of the
codebuild-RWBCore-managed-policy policy that is attached to the codebuild-
RWBCore-service-role role. To continue, detach the policy from any other
identities and then delete the policy and the role.
```

Se viene visualizzato questo errore, dovrai apportare le modifiche in IAM prima di poter continuare con l'operazione di servizio. Innanzitutto, impostare la versione predefinita del criterio su V1 e

riprovare l'operazione. Se V1 è stato eliminato in precedenza o se la scelta di V1 non funziona, pulire ed eliminare il criterio e il ruolo esistenti.

Per ulteriori informazioni sulla modifica dei criteri gestiti, vedere Modifica di policy gestite dal cliente (console). Per ulteriori informazioni sulle versioni dei criteri, consulta Controllo delle versioni delle policy IAM.

Per eliminare un ruolo di servizio e il relativo criterio

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Nell'elenco delle policy, selezionare il nome della policy che si desidera modificare.
4. Scegli la scheda Entità collegate per visualizzare gli utenti, i gruppi o i ruoli IAM che utilizzano questa policy. Se una di queste identità utilizza il criterio, completare le seguenti attività:
    a. Creare un nuovo criterio gestito con le autorizzazioni necessarie. Per assicurarsi che le identità dispongano delle stesse autorizzazioni prima e dopo le azioni, copiare il documento dei criteri JSON dal criterio esistente. Crea quindi la nuova policy gestita e incolla il documento JSON come descritto in Creazione di policy utilizzando l'editor JSON.
    b. Per ogni identità interessata, allegare il nuovo criterio e quindi staccare quello precedente. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.
5. Nel riquadro di navigazione, seleziona Ruoli.
6. Nell'elenco dei ruoli scegliere il nome del ruolo che si desidera eliminare.
7. Scegliere la scheda Relazioni di trust per visualizzare le entità che possono assumere il ruolo. Se è elencata un'entità diversa dal servizio, completare le seguenti attività:
    a. Creare un nuovo ruolo che si attenda a tali entità.
    b. Pertanto, devi collegare la policy creata nella fase precedente. Se questo passaggio è stato ignorato, creare subito il nuovo criterio gestito.
    c. Informare chiunque abbia assunto il ruolo che non può più farlo. Fornire loro informazioni su come assumere il nuovo ruolo e disporre delle stesse autorizzazioni.
8. Eliminare il criterio.
9. Eliminare il ruolo.

Non esiste un caso d'uso per un ruolo di servizio nella console

Alcuni servizi richiedono la creazione manuale di un ruolo di servizio per concedere al servizio autorizzazioni per eseguire operazioni per conto dell'utente. Se il servizio non è elencato nella console IAM, dovrai elencarlo manualmente come principale attendibile. Se la documentazione relativa al servizio o alla funzionalità in uso non include istruzioni per elencare il servizio come principale attendibile, fornisci un feedback sulla pagina.

Per creare manualmente un ruolo di servizio, è necessario conoscere il principale del servizio per il servizio che assumerà il ruolo. Un'entità servizio è un identificatore che viene utilizzato per concedere autorizzazioni a un servizio. Il principale del servizio è definito dal servizio.

È possibile trovare il principale del servizio per alcuni servizi con la seguente procedura:

1. Aprire AWS servizi che funzionano con IAM.
2. Verificare se per il servizio è indicato Sì nella colonna Ruoli collegati ai servizi .
3. Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.
4. Trova la sezione Autorizzazioni del ruolo collegato ai servizi per quel servizio per visualizzare il principale del servizio.

È possibile creare manualmente un ruolo del servizio utilizzando i comandi della AWS CLI o le operazioni delle API AWS. Per creare manualmente un ruolo di servizio utilizzando la console IAM, completa le seguenti attività:

1. Crea un ruolo IAM utilizzando il tuo ID account. Non allegare una policy o concedere autorizzazioni. Per informazioni dettagliate, vedi Creazione di un ruolo per delegare le autorizzazioni a un utente IAM.
2. Apri il ruolo e modificare la relazione di attendibilità. Invece di fidarsi dell'account, il ruolo deve considerare attendibile il servizio. Ad esempio, aggiorna il seguente elemento `Principal`:

   ```
   "Principal": { "AWS": "arn:aws:iam::123456789012:root" }
   ```

   Cambia il valore del principale per il servizio, ad esempio IAM.

   ```
   "Principal": { "Service": "iam.amazonaws.com" }
   ```
3. Aggiungi le autorizzazioni richieste dal servizio allegando le policy di autorizzazione al ruolo.

4. Torna al servizio che richiede le autorizzazioni e utilizza il metodo documentato per notificare al servizio il nuovo ruolo di servizio.

Risoluzione dei problemi relativi a IAM e Amazon EC2

Utilizza le informazioni contenute in questa pagina per risolvere i problemi di accesso negato e di altro tipo che possono verificarsi durante l'utilizzo di Amazon EC2 e IAM.

Argomenti

Durante l'avvio di un'istanza, non viene visualizzato il ruolo previsto nell'elenco Ruolo IAM nella console Amazon EC2.

Verifica quanto segue:

- Se si è effettuato l'accesso come utente IAM, verificare di disporre dell'autorizzazione a chiamare `ListInstanceProfiles`. Per informazioni sulle autorizzazioni necessarie per lavorare con i ruoli, consulta "Autorizzazioni richieste per l'utilizzo dei ruoli con Amazon EC2" in Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestione di policy IAM.

  Se non puoi modificare le tue autorizzazioni, contatta un amministratore che possa utilizzare IAM per aggiornare le autorizzazioni.
- Se è stato creato un ruolo utilizzando la CLI o l'API di IAM, verifica di avere creato un profilo dell'istanza e di avere aggiunto il ruolo a tale profilo. Inoltre, se il ruolo e il profilo dell'istanza sono stati chiamati in modo diverso, il nome del ruolo corretto non verrà visualizzato nell'elenco di ruoli IAM nella console Amazon EC2. L'elenco Ruolo IAM nella console Amazon EC2 riporta i nomi dei profili dell'istanza, non i nomi dei ruoli. Sarà necessario selezionare il nome del profilo dell'istanza

che contiene il ruolo desiderato. Per ulteriori informazioni sui profili dell'istanza, consultare Utilizzo dei profili delle istanze.

> **Note**
>
> Se utilizzi la console IAM per creare ruoli, non è necessario lavorare con i profili dell'istanza. Per ogni ruolo creato nella console IAM viene creato un profilo dell'istanza con lo stesso nome del ruolo e il ruolo viene automaticamente aggiunto a tale profilo. Un profilo di istanza può contenere un solo ruolo IAM e tale limite non può essere aumentato.

Le credenziali per l'istanza si riferiscono al ruolo errato

Il ruolo nel profilo dell'istanza potrebbe essere stato sostituito di recente. In questo caso, l'applicazione deve attendere la prossima rotazione delle credenziali pianificata automaticamente prima che le credenziali per il ruolo diventino disponibili.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Quando tento di chiamare **`AddRoleToInstanceProfile`**, viene visualizzato un errore **`AccessDenied`**

Se stai effettuando richieste come utente IAM, verifica di disporre delle autorizzazioni seguenti:

- `iam:AddRoleToInstanceProfile` con la risorsa corrispondente all'ARN del profilo dell'istanza (ad esempio, `arn:aws:iam::999999999999:instance-profile/ExampleInstanceProfile`).

Per ulteriori informazioni sulle autorizzazioni necessarie per iniziare a utilizzare i ruoli, consulta "Come inizio?" nella sezione Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestione di policy IAM.

Amazon EC2: quando provo ad avviare un'istanza con un ruolo, ricevo un errore **`AccessDenied`**.

Verifica quanto segue:

- Avviare un'istanza senza un profilo dell'istanza. In questo modo il problema sarà limitato ai ruoli IAM per le istanze Amazon EC2.
- Se stai effettuando richieste come utente IAM, verifica di disporre delle autorizzazioni seguenti:
  - `ec2:RunInstances` con una risorsa jolly ("*")
  - `iam:PassRole` con la risorsa corrispondente all'ARN del ruolo (ad esempio, `arn:aws:iam::999999999999:role/ExampleRoleName`)
- Chiama l'operazione `GetInstanceProfile` IAM per assicurarti di stare utilizzando un profilo dell'istanza valido o un ARN del profilo di istanza valido. Per ulteriori informazioni, consulta Utilizzo dei ruoli IAM con le istanze Amazon EC2.
- Chiama l'operazione `GetInstanceProfile` IAM per assicurarti che il profilo di istanza disponga di un ruolo. I profili dell'istanza vuoti genereranno un errore `AccessDenied`. Per ulteriori informazioni sulla creazione di un ruolo, consultare Creazione di ruoli IAM.

Per ulteriori informazioni sulle autorizzazioni necessarie per iniziare a utilizzare i ruoli, consulta "Come inizio?" nella sezione Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2. Per informazioni sull'aggiunta di autorizzazioni a un utente, consultare Gestione di policy IAM.

Non è possibile accedere alle credenziali di sicurezza temporanee nell'istanza EC2

Per accedere alle credenziali di sicurezza temporanee nell'istanza EC2, è necessario innanzitutto utilizzare la console IAM per creare un ruolo. Quindi, avviare un'istanza EC2 che utilizza tale ruolo ed esaminare l'istanza in esecuzione. Per ulteriori informazioni, consulta How Do I Get Started? in Utilizzo di un ruolo IAM per concedere autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Se non è ancora possibile accedere alle credenziali di sicurezza temporanee sull'istanza EC2, verificare quanto segue:

- È possibile accedere a un'altra parte di Instance Metadata Service (IMDS)? In caso contrario, verificare che non vi siano regole del firewall che bloccano l'accesso alle richieste a IMDS.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/
hostname; echo
```

- La sottostruttura `iam` di IMDS esiste? In caso contrario, verifica che all'istanza sia associato un profilo dell'istanza IAM chiamando l'operazione API `DescribeInstances` di EC2 o utilizzando il comando della CLI `aws ec2 describe-instances`.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam;
 echo
```

- Verifica la presenza di un errore nel documento `info` nella sottostruttura IAM. Se è presente un errore, consultare Cosa significano gli errori riportati nel documento `info` nella sottostruttura IAM? per ulteriori informazioni.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam/
info; echo
```

Cosa significano gli errori riportati nel documento **`info`** nella sottostruttura IAM?

Il documento **`iam/info`** indica **`"Code":"InstanceProfileNotFound"`**

Il profilo dell'istanza IAM è stato eliminato e Amazon EC2 non è più in grado di fornire le credenziali all'istanza. È necessario collegare un profilo dell'istanza valido all'istanza Amazon EC2.

Se esiste un profilo dell'istanza con il nome specificato, controllare che tale profilo non sia stato eliminato e che ne sia stato creato un altro con lo stesso nome:

1. Chiama l'operazione `GetInstanceProfile` IAM per ottenere `InstanceProfileId`.
2. Chiama l'operazione `DescribeInstances` di Amazon EC2 per ottenere il valore `IamInstanceProfileId` per l'istanza.
3. Verifica che il `InstanceProfileId` ottenuto dall'operazione IAM corrisponda al `IamInstanceProfileId` ottenuto dall'operazione Amazon EC2.

Se gli ID sono diversi, il profilo dell'istanza associato alle istanze non è più valido. È necessario collegare un profilo dell'istanza valido all'istanza.

Il documento **iam/info** indica un esito positivo, ma indica anche **"Message":"Instance Profile does not contain a role..."**

Il ruolo è stato rimosso dal profilo dell'istanza dall'operazione `RemoveRoleFromInstanceProfile` IAM. È possibile utilizzare l'operazione `AddRoleToInstanceProfile` IAM per collegare un ruolo al profilo dell'istanza. L'applicazione dovrà attendere il successivo aggiornamento pianificato per accedere alle credenziali del ruolo.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Il documento **iam/security-credentials/[role-name]** indica **"Code":"AssumeRoleUnauthorizedAccess"**

Amazon EC2 non dispone dell'autorizzazione per assumere il ruolo. L'autorizzazione ad assumere il ruolo è determinata dalla policy di affidabilità collegata al ruolo, come nell'esempio che segue. Utilizza l'API `UpdateAssumeRolePolicy` IAM per aggiornare la policy di attendibilità.

```
{"Version": "2012-10-17","Statement": [{"Effect": "Allow","Principal": {"Service":
 ["ec2.amazonaws.com"]},"Action": ["sts:AssumeRole"]}]}
```

L'applicazione dovrà attendere il successivo aggiornamento pianificato automaticamente per accedere alle credenziali del ruolo.

Per forzare la modifica, devi dissociare il profilo dell'istanza e quindi associare il profilo dell'istanza oppure arrestare l'istanza e riavviarla.

Risoluzione dei problemi di IAM ed Amazon S3

Utilizza le informazioni contenute in questa pagina per eseguire la diagnosi e risolvere i problemi che possono verificarsi durante l'utilizzo di Amazon S3 e IAM.

Come posso concedere l'accesso anonimo a un bucket Amazon S3?

È possibile utilizzare una policy del bucket Amazon S3 che specifica un carattere jolly (*) nell'elemento `principal`, il che significa che chiunque è in grado di accedere al bucket. Con l'accesso anonimo, chiunque (compresi gli utenti senza un Account AWS) potrà accedere al bucket.

Per una policy di esempio, consulta Esempi di policy di bucket Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Ho effettuato l'accesso come utente Account AWS root; perché non riesco ad accedere a un bucket Amazon S3 dal mio account?

In alcuni casi, è possibile avere un utente IAM con accesso completo a IAM e Amazon S3. Se l'utente IAM assegna una policy bucket a un bucket Amazon S3 e non Utente root dell'account AWS la specifica come principale, all'utente root viene negato l'accesso a quel bucket. Tuttavia, come utente root, puoi comunque accedere al bucket. A tale scopo, modifica la policy del bucket per consentire l'accesso dell'utente root dalla console Amazon S3 o la AWS CLI. Utilizza il seguente principale, sostituendo *123456789012* con l'ID dell' Account AWS.

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

Risoluzione dei problemi di federazione SAML 2.0 con AWS

Utilizza le informazioni contenute qui per eseguire la diagnosi e risolvere i problemi che puoi incontrare durante l'utilizzo di SAML 2.0 e la federazione con IAM.

Argomenti

- Errore: La tua richiesta include una risposta SAML non valida. Per disconnetterti, fai clic qui.
- Errore: RoleSessionName è richiesto in AuthnResponse (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)
- Errore: non autorizzato a eseguire sts: AssumeRole withSAML (service: AWSSecurityTokenService; codice di stato: 403; codice di errore:) AccessDenied
- Errore: RoleSessionName in AuthnResponse deve corrispondere a [a-zA-Z_0-9+=, .@-] {2,64} (service:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken
- Errore: l'identità di origine deve corrispondere a [a-zA-Z_0-9+=, .@-] {2,64} e non iniziare con "aws:" (service:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken
- Errore: firma di risposta non valida (servizio:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken
- Errore: impossibile assumere il ruolo: Emittente non presente nel provider specificato (servizio:; codice di stato: 400; codice di errore: AWSOpenIdDiscoveryService) AuthSamlInvalidSamlResponseException

- Errore: Impossibile analizzare i metadati.
- Errore: Il provider specificato non esiste.
- Errore: la richiesta DurationSeconds supera quella MaxSessionDuration impostata per questo ruolo.
- Errore: la risposta non contiene il pubblico richiesto.
- Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi

Errore: La tua richiesta include una risposta SAML non valida. Per disconnetterti, fai clic qui.

Questo errore può accadere quando la risposta SAML dall'identità del fornitore non include un attributo con `Name` impostato su `https://aws.amazon.com/SAML/Attributes/Role` L'attributo deve contenere uno o più elementi `AttributeValue`, ognuno contenente un paio di stringhe separate dalla virgola:

- L'ARN di un ruolo su cui l'utente può essere mappato
- L'ARN del fornitore SAML

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi.

Errore: RoleSessionName è richiesto in AuthnResponse (servizio: AWSSecurityTokenService; codice di stato: 400; codice di errore: InvalidIdentityToken)

Questo errore può accadere quando la risposta SAML dall'identità del fornitore non include un attributo con `Name` impostato su `https://aws.amazon.com/SAML/Attributes/RoleSessionName` Il valore dell'attributo è un identificatore per l'utente e in genere è un ID utente o un indirizzo e-mail.

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi.

Errore: non autorizzato a eseguire sts: AssumeRole withSAML (service: AWSSecurityTokenService; codice di stato: 403; codice di errore:) AccessDenied

Questo errore può verificarsi se il ruolo IAM specificato nella risposta SAML è errato o non esiste. Assicurati di utilizzare il nome esatto del ruolo in quanto i nomi prevedono una distinzione tra lettere maiuscole e minuscole. Correggere il nome del ruolo nella configurazione del provider di servizi SAML.

L'accesso è consentito solo se il criterio di attendibilità del ruolo include l'azione `sts:AssumeRoleWithSAML`. Se l'asserzione SAML è configurata per utilizzare l'attributo `PrincipalTag`, i criteri di attendibilità devono includere anche l'azione `sts:TagSession`. Per ulteriori informazioni sui tag di sessione, consultare Passare i tag di sessione AWS STS.

Questo errore può verificarsi se non disponi delle autorizzazioni `sts:SetSourceIdentity` nella policy di attendibilità del ruolo. Se l'asserzione SAML è configurata per utilizzare l'attributo `SourceIdentity`, le policy di attendibilità devono includere anche l'azione `sts:SetSourceIdentity`. Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Questo errore può verificarsi se gli utenti federati non hanno le autorizzazioni per assumere quel ruolo. Il ruolo deve avere una policy di affidabilità che specifica l'ARN del provider d'identità SAML IAM come il `Principal`. Il ruolo contiene anche le condizioni che controllano quali utenti possono assumere il ruolo. Verifica che gli utenti soddisfino i requisiti delle condizioni.

Questo errore può verificarsi anche se la risposta SAML non include un `Subject` contenente un `NameID`.

Per ulteriori informazioni, consulta Come stabilire le autorizzazioni in AWS per gli utenti federati e Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi.

Errore: RoleSessionName in AuthnResponse deve corrispondere a [a-zA-Z_0-9+=, .@-] {2,64} (service:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken

Questo errore può verificarsi se il valore dell'attributo `RoleSessionName` è troppo lungo o contiene caratteri non validi. La lunghezza massima valida è 64 caratteri;

Per ulteriori informazioni, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi.

Errore: l'identità di origine deve corrispondere a [a-zA-Z_0-9+=, .@-] {2,64} e non iniziare con **`"aws:"`** (service:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken

Questo errore può verificarsi se il valore dell'attributo `sourceIdentity` è troppo lungo o contiene caratteri non validi. La lunghezza massima valida è 64 caratteri; Per ulteriori informazioni sull'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

Per ulteriori informazioni sulla creazione di asserzioni SAML, consulta Configurare le asserzioni SAML per la risposta di autenticazione. Per visualizzare la risposta SAML nel browser, seguire le fasi elencate in Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi.

Errore: firma di risposta non valida (servizio:; codice di stato: 400; codice di errore:) AWSSecurityTokenService InvalidIdentityToken

Questo errore può verificarsi quando i metadati di federazione del provider di identità non soddisfano i metadati del provider di identità IAM. Ad esempio, il file dei metadati per il provider del servizio di identità potrebbe essere cambiato per aggiornare un certificato scaduto. Scaricare il file di metadati SAML aggiornato dal fornitore del servizio di identità. Quindi aggiornalo nell'entità del provider di AWS identità che definisci in IAM con il comando CLI `aws iam update-saml-provider` multipiattaforma o `Update-IAMSAMLProvider` PowerShell il cmdlet.

Errore: impossibile assumere il ruolo: Emittente non presente nel provider specificato (servizio:; codice di stato: 400; codice di errore: AWSOpenIdDiscoveryService) AuthSamlInvalidSamlResponseException

Questo errore può verificarsi se l'approvatore nella risposta SAML non corrisponde all'approvatore dichiarato nel file dei metadati di federazione Il file di metadati è stato caricato su AWS quando hai creato il provider di identità in IAM.

Errore: Impossibile analizzare i metadati.

Questo errore può verificarsi se il file dei metadati non è formattato correttamente.

Quando crei o gestisci un provider di identità SAML in AWS Management Console, devi recuperare il documento di metadati SAML dal tuo provider di identità.

Questo file di metadati include il nome dell'approvatore, le informazioni sulla scadenza e le chiavi che possono essere utilizzate per convalidare la risposta di autenticazione SAML (asserzioni) ricevute dal provider di identità. Il file di metadati deve essere codificati in formato UTF-8, senza BOM (Byte Order Mark). Per rimuovere il BOM, codifica i file come UTF-8 utilizzando un editor di testi come ad esempio Notepad++.

Il certificato x.509 incluso come parte del documento di metadati SAML deve utilizzare una chiave di almeno 1024 bit. Inoltre, il certificato x.509 deve essere privo di eventuali estensioni ripetute. È possibile utilizzare le estensioni, ma possono essere visualizzate una sola volta nel certificato. Se il certificato x.509 non soddisfa nessuna delle due condizioni, la creazione dell'IdP ha esito negativo e restituisce l'errore "Unable to parse metadata".

Come definito dal profilo di interoperabilità dei metadati SAML V2.0 versione 1.0, IAM non valuta né interviene in merito alla scadenza del certificato X.509 del documento di metadati.

Errore: Il provider specificato non esiste.

Questo errore può verificarsi se il nome del provider specificato nell'asserzione SAML non corrisponde al nome del provider configurato in IAM. Per ulteriori informazioni sulla visualizzazione del nome del provider, consulta Crea un provider di identità SAML in IAM.

Errore: la richiesta DurationSeconds supera quella MaxSessionDuration impostata per questo ruolo.

Questo errore può verificarsi se assumi un ruolo dall'API AWS CLI o.

Quando utilizzi le operazioni dell'interfaccia della riga di comando assume-role-with-saml AssumeRole o dell'API WithSAML per assumere un ruolo, puoi specificare un valore per il parametro. `DurationSeconds` Puoi specificare un valore da 900 secondi (15 minuti) fino alla durata massima impostata per la sessione per il ruolo. Se si specifica un valore superiore a questa impostazione, l'operazione ha esito negativo. Ad esempio, se si specifica una durata di sessione di 12 ore, ma l'amministratore ha impostato la durata massima di sessione a 6 ore, l'operazione ha esito negativo. Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.

Errore: la risposta non contiene il pubblico richiesto.

Questo errore può verificarsi in caso di mancata corrispondenza tra l'URL del pubblico e il provider di identità nella configurazione SAML. Assicurati che l'identificativo del soggetto che si basa sul gestore dell'identità digitale corrisponda esattamente all'URL del pubblico (ID entità) fornito nella configurazione SAML.

Come visualizzare una risposta SAML nel browser per la risoluzione dei problemi

Le procedure seguenti descrivono come visualizzare nel browser la risposta SAML del proprio provider di servizi durante la risoluzione di un problema relativo a SAML 2.0.

Per tutti i browser, passare alla pagina in cui è possibile riprodurre il problema. Quindi seguire i passaggi per il browser appropriato:

Argomenti

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Operazioni da effettuare con la risposta SAML codificata Base64

Google Chrome

Per visualizzare una risposta SAML in Chrome

Questi passaggi sono stati testati utilizzando la versione 106.0.5249.103 (versione ufficiale) (arm64) di Google Chrome. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Premere F12 per avviare la console Strumenti per sviluppatori.
2. Selezionare la scheda Netowrk (Rete), quindi selezionare Preserve log (Conserva registro) nella parte superiore sinistra della finestra Developer Tools (Strumenti per sviluppatori).
3. Riprodurre il problema.
4. (Facoltativo) Se la colonna Method (Metodo) non è visibile nel pannello di registrazione Developer Tools (Strumenti per sviluppatori) Network (Rete), fare clic con il pulsante destro del mouse su qualsiasi etichetta di colonna e scegliere Method (Metodo) per aggiungere la colonna
5. Cercare un SAML Post (Post SAML) nel pannello di registrazione Developer Tools (Strumenti per sviluppatori) Network (Rete). Selezionare la riga e quindi visualizzare la scheda Payload (Carico utile) nella parte superiore. Cercare l'elemento SAMLResponse che contiene la richiesta codificata. Il valore associato è la risposta codificata Base64.

Mozilla Firefox

Per visualizzare una risposta SAML in Firefox

Questa procedura è stata testata con la versione 105.0.3 (64 bit) di Mozilla Firefox. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Premere F12 per avviare la console Strumenti per sviluppatori Web.
2. Selezionare la scheda Network (Rete).
3. In alto a destra nella finestra Web Developer Tools (Strumenti per sviluppatori Web), fare clic sull'icona delle opzioni (il piccolo ingranaggio). Selezionare Persist logs (Preserva registri).
4. Riprodurre il problema.
5. (Facoltativo) Se la colonna Method (Metodo) non è visibile nel pannello di registrazione Developer Tools (Strumenti per sviluppatori Web) Network (Rete), fare clic con il pulsante destro del mouse su qualsiasi etichetta di colonna e scegliere Method (Metodo) per aggiungere la colonna

6. Cercare un MESSAGGIO SAML nella tabella. Selezionare la riga e quindi visualizzare la scheda Request (Richiesta) e trovare l'elemento SAMLResponse. Il valore associato è la risposta codificata Base64.

Apple Safari

Per visualizzare una risposta Safari

Questi passaggi sono stati testati utilizzando la versione 16.0 (17614.1.25.9.10, 17614) di Apple Safari. Se si utilizza una versione diversa, potrebbe essere necessario modificare i passaggi di conseguenza.

1. Abilitare Web Inspector in Safari. Aprire la finestra delle preferenze selezionare la scheda delle impostazioni avanzate e quindi selezionare l'opzione per mostrare il menu Sviluppo nella barra dei menu.
2. Ora è possibile aprire Web Inspector. Scegliere Develop (Sviluppo) nella barra dei menu, quindi selezionare Show Web Inspector (Mostra Web Inspector).
3. Selezionare la scheda Network (Rete).
4. Nella parte superiore sinistra della finestra Web Inspector, fare clic sull'icona delle opzioni (il piccolo cerchio con tre linee orizzontali). Selezionare Preserve Logs (Conserva registri).
5. (Facoltativo) Se la colonna Method (Metodo) non è visibile nel pannello di registrazione Web Inspector Network (Rete), fare clic con il pulsante destro del mouse su qualsiasi etichetta di colonna e scegliere Method (Metodo) per aggiungere la colonna
6. Riprodurre il problema.
7. Cercare un MESSAGGIO SAML nella tabella. Selezionare la riga e quindi visualizzare la scheda Headers (Intestazioni).
8. Cercare l'elemento SAMLResponse che contiene la richiesta codificata. Scorrere per trovare l'elemento `Request Data` con nome `SAMLResponse`. Il valore associato è la risposta codificata Base64.

Operazioni da effettuare con la risposta SAML codificata Base64

Una volta trovato l'elemento di risposta SAML con codifica Base64 nel browser, copiarlo e utilizzare lo strumento di decodifica Base-64 preferito per estrarre la risposta con tag XML.

> **Suggerimento per la sicurezza**
>
> Poiché i dati di risposta SAML visualizzati potrebbero contenere dati di sicurezza sensibili, si consiglia di non utilizzare un decodificatore base64. Utilizzare invece uno strumento installato sul computer locale che non invia i dati SAML sulla rete.

Opzione integrata per sistemi Windows (PowerShell):

```
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("base64encodedtext
```

Opzione integrata per sistemi MacOS e Linux:

```
$ echo "base64encodedtext" | base64 --decode
```

Rivedi i valori nel file decodificato

Rivedi i valori nel file di risposta SAML decodificato.

- Verifica che il valore dell'attributo SAML:NameID corrisponda al nome utente dell'utente autenticato.
- Controlla il valore per https://aws.amazon.com/SAML/Attributes/Role. I provider ARN e SAML fanno distinzione tra maiuscole e minuscole e l'ARN deve corrispondere alla risorsa del tuo account.
- Controlla il valore per https://aws.amazon.com/SAML/Attributes/ Name. RoleSession Il valore deve corrispondere al valore indicato nella regola di reclamo.
- Se configuri il valore dell'attributo per un indirizzo e-mail o un nome di account, assicurati che i valori siano corretti. I valori devono corrispondere all'indirizzo e-mail o al nome dell'account dell'utente autenticato.

Verifica la presenza di errori e conferma la configurazione

Controlla se i valori contengono errori e conferma che le seguenti configurazioni siano corrette.

- Le regole di reclamo soddisfano gli elementi richiesti e tutti gli ARN sono corretti. Per ulteriori informazioni, consulta Configura il tuo IdP SAML 2.0 con la fiducia dei relying party e l'aggiunta di claim.

- Hai caricato il file di metadati più recente dal tuo IdP AWS nel tuo provider SAML. Per ulteriori informazioni, consulta Consentire agli utenti federati SAML 2.0 di accedere a AWS Management Console.
- Hai configurato correttamente la policy di fiducia del ruolo IAM. Per ulteriori informazioni, consulta Modifica di un ruolo.

Informazioni di riferimento per AWS Identity and Access Management

Utilizza gli argomenti di questa sezione per trovare materiali di riferimento dettagliati su diversi aspetti di IAM e AWS STS.

Argomenti

Amazon Resource Names (ARN)

Amazon Resource Names (ARN) identifica AWS in modo univoco le risorse. Abbiamo bisogno di un ARN quando è necessario specificare una risorsa in modo inequivocabile per tutti AWS, ad esempio nelle policy IAM, nei tag Amazon Relational Database Service (Amazon RDS) e nelle chiamate API.

Formato ARN

Di seguito sono riportati i formati generali per gli ARN. I formati specifici dipendono dalla risorsa. Per utilizzare un ARN, sostituire il testo *in corsivo* con le informazioni specifiche delle risorse. Tenere presente che gli ARN per alcune risorse omettono la regione, l'ID account o la regione e l'ID account.

```
arn:partition:service:region:account-id:resource-id
arn:partition:service:region:account-id:resource-type/resource-id
arn:partition:service:region:account-id:resource-type:resource-id
```

partition

La partizione in cui si trova la risorsa. Una partizione è un gruppo di regioni. AWS Ogni AWS account è limitato a una partizione.

Di seguito sono riportate le partizioni supportate:

- `aws`- Regioni AWS
- `aws-cn`: regioni Cina
- `aws-us-gov` – Regioni AWS GovCloud (US)

`service`

Lo spazio dei nomi del servizio che identifica il prodotto. AWS

`region`

Il codice della regione. Ad esempio, `us-east-2` per Stati Uniti orientali (Ohio). Per un elenco dei codici delle regioni, consulta Endpoint regionali nella Riferimenti generali di AWS.

`account-id`

L'ID dell' AWS account proprietario della risorsa, senza i trattini. Ad esempio, `123456789012`.

`resource-type`

Il tipo di risorsa. Ad esempio, `vpc` per un cloud privato virtuale (VPC).

`resource-id`

L'identificatore di risorsa. Si tratta del nome della risorsa, dell'ID della risorsa o del percorso della risorsa. Alcuni identificatori di risorse includono una risorsa principale (sub-resource-type/parent-resource/sub-resource) o un qualificatore come una versione (resource-type:resource-name:qualifier).

Esempi

Utente IAM

arn:aws:iam::*`123456789012`*:user/*`johndoe`*

Argomento SNS

`arn:aws:sns:us-east-1:123456789012: example-sns-topic-name`

VPC

arn:aws:ec2:*`us-east-1`*:*`123456789012`*:vpc/*`vpc-0e9801d129EXAMPLE`*

Ricerca del formato dell'ARN per una risorsa

Il formato esatto di un ARN dipende dal servizio e dal tipo di risorsa. Alcuni ARN di risorse possono includere un percorso, una variabile o un carattere jolly. Per cercare il formato ARN per una AWS risorsa specifica, apri il Service Authorization Reference, apri la pagina del servizio e vai alla tabella dei tipi di risorse.

Percorsi negli ARN

Alcuni ARN delle risorse possono includere un percorso. Ad esempio, in Amazon S3, l'identificatore di risorsa è un nome oggetto che può includere barre in avanti (`/`) per creare un percorso. Allo stesso modo, anche i nomi utente e i nomi di gruppo IAM possono includere percorsi. Nei percorsi IAM sono consentiti solo caratteri alfanumerici e i seguenti caratteri: barra obliqua (`/`), segno del più (`+`), segno dell'uguale (`=`), virgola (`,`), punto (`.`) chiocciola (`@`) trattino basso (`_`) e trattino (`-`).

Utilizzo di caratteri jolly nei percorsi

I percorsi possono includere un carattere jolly, vale a dire un asterisco (`*`). Ad esempio, se si sta scrivendo una policy IAM, è possibile specificare tutti gli utenti IAM che dispongono del percorso `product_1234` utilizzando un carattere jolly come segue:

```
arn:aws:iam::123456789012:user/Development/product_1234/*
```

Analogamente, puoi specificare `user/*` per indicare tutti gli utenti o `group/*` per indicare tutti i gruppi, come negli esempi seguenti:

```
"Resource":"arn:aws:iam::123456789012:user/*"
"Resource":"arn:aws:iam::123456789012:group/*"
```

L'esempio seguente mostra gli ARN per un bucket Amazon S3 in cui il nome della risorsa include un percorso:

```
arn:aws:s3:::my_corporate_bucket/*
arn:aws:s3:::my_corporate_bucket/Development/*
```

Utilizzo non corretto dei caratteri jolly

Non è possibile utilizzare un carattere jolly nella parte di ARN che specifica il tipo di risorsa, ad esempio il termine `user` in un ARN IAM. Ad esempio, non è consentito quanto segue:

```
arn:aws:iam::123456789012:u*   <== not allowed
```

Identificatori IAM

IAM utilizza vari identificatori per utenti, gruppi di utenti, ruoli, policy e certificati del server. Questa sezione descrive gli identificatori e spiega quando vanno utilizzati.

Argomenti

Nomi descrittivi e percorsi

Quando crei un utente, un ruolo, un gruppo di utenti o una policy o quando carichi un certificato server, gli attribuisci un nome descrittivo. Gli esempi includono Bob, TestApp 1, Developers ManageCredentialsPermissions, o ProdServerCert.

Se utilizzi l'API IAM o AWS Command Line Interface (AWS CLI) per creare risorse IAM, puoi aggiungere un percorso opzionale. Puoi decidere di usare un solo percorso o nidificare percorsi multipli come una struttura a cartelle. Ad esempio, puoi utilizzare il percorso nidificato `/division_abc/subdivision_xyz/product_1234/engineering/` per farlo corrispondere alla struttura organizzativa della tua azienda. A quel punto, potresti ceare una policy per consentire a tutti gli utenti del percorso di accedere all'API di simulazione policy. Per visualizzare questa policy, consulta IAM: accesso all'API simulatore di policy basata sul percorso degli utenti. Per informazioni su come specificare un nome descrittivo, vedere la documentazione dell'API utente. Per ulteriori esempi di utilizzo dei percorsi, consulta ARN IAM.

Quando utilizzi AWS CloudFormation per creare risorse, puoi specificare un percorso per utenti, gruppi di utenti e ruoli e politiche gestite dai clienti.

Se disponi di un utente e di un gruppo di utenti nello stesso percorso, IAM non inserisce automaticamente l'utente in tale gruppo di utenti. Ad esempio, potresti creare il gruppo Sviluppatori e specificare come percorso come `/division_abc/subdivision_xyz/product_1234/engineering/`. Se crei un utente denominato Bob e gli aggiungi lo stesso percorso, Bob non viene inserito automaticamente all'interno del gruppo di utenti Sviluppatori. IAM non impone limiti tra utenti o gruppi di utenti in base ai loro percorsi. Utenti con percorsi differenti possono utilizzare le stesse

risorse (supponendo che abbiano ricevuto le autorizzazioni per farlo). Il numero e la dimensione delle risorse IAM in un AWS account sono limitati. Per ulteriori informazioni, consulta IAM e AWS STS quote.

ARN IAM

La maggior parte delle risorse dispone di nomi descrittivi: ad esempio, un utente denominato `Bob` o un gruppo di utenti denominato `Developers`. Tuttavia, il linguaggio delle policy di autorizzazione richiede di specificare la risorsa o le risorse che utilizzano il seguente formato Amazon Resource Name (ARN).

```
arn:partition:service:region:account:resource
```

Dove:

- `partition` identifica la partizione in cui si trova la risorsa. Per le Regioni AWS standard, la partizione è `aws`. Se sono presenti risorse in altre partizioni, la partizione è `aws-partitionname`. Ad esempio, la partizione per le risorse nella regione Cina (Pechino) è `aws-cn`. Non è possibile delegare l'accesso tra account in partizioni diverse.
- `service`identifica il AWS prodotto. Le risorse IAM utilizzano sempre `iam`.
- `region` identifica la regione della risorsa. Per le risorse IAM, è sempre lasciato vuoto.
- `account`specifica l' Account AWS ID senza trattini.
- `resource` identifica la risorsa specifica in base al nome.

È possibile specificare IAM e AWS STS ARN utilizzando la seguente sintassi. La porzione di regione dell'ARN è vuota perché le risorse IAM sono globali.

Sintassi:

```
arn:aws:iam::account:root
arn:aws:iam::account:user/user-name-with-path
arn:aws:iam::account:group/group-name-with-path
arn:aws:iam::account:role/role-name-with-path
arn:aws:iam::account:policy/policy-name-with-path
arn:aws:iam::account:instance-profile/instance-profile-name-with-path
arn:aws:sts::account:federated-user/user-name
arn:aws:sts::account:assumed-role/role-name/role-session-name
arn:aws:sts::account:self
```

```
arn:aws:iam::account:mfa/virtual-device-name-with-path
arn:aws:iam::account:u2f/u2f-token-id
arn:aws:iam::account:server-certificate/certificate-name-with-path
arn:aws:iam::account:saml-provider/provider-name
arn:aws:iam::account:oidc-provider/provider-name
```

Molti esempi riportati di seguito includono percorsi nella parte della risorsa dell'ARN. I percorsi non possono essere creati o modificati nella AWS Management Console. Per utilizzare i percorsi, è necessario utilizzare la risorsa utilizzando l' AWS API AWS CLI, il o gli strumenti per Windows. PowerShell

Esempi:

```
arn:aws:iam::123456789012:root
arn:aws:iam::123456789012:user/JohnDoe
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/JaneDoe
arn:aws:iam::123456789012:group/Developers
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
arn:aws:iam::123456789012:role/S3Access
arn:aws:iam::123456789012:role/application_abc/component_xyz/RDSAccess
arn:aws:iam::123456789012:role/aws-service-role/access-analyzer.amazonaws.com/
AWSServiceRoleForAccessAnalyzer
arn:aws:iam::123456789012:role/service-role/QuickSightAction
arn:aws:iam::123456789012:policy/UsersManageOwnCredentials
arn:aws:iam::123456789012:policy/division_abc/subdivision_xyz/UsersManageOwnCredentials
arn:aws:iam::123456789012:instance-profile/Webserver
arn:aws:sts::123456789012:federated-user/JohnDoe
arn:aws:sts::123456789012:assumed-role/Accounting-Role/JaneDoe
arn:aws:sts::123456789012:self
arn:aws:iam::123456789012:mfa/JaneDoeMFA
arn:aws:iam::123456789012:u2f/user/JohnDoe/default (U2F security key)
arn:aws:iam::123456789012:server-certificate/ProdServerCert
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/
ProdServerCert
arn:aws:iam::123456789012:saml-provider/ADFSProvider
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-west-2.amazonaws.com/id/
a1b2c3d4567890abcdefEXAMPLE11111
arn:aws:iam::123456789012:oidc-provider/server.example.org
```

Gli esempi seguenti forniscono maggiori dettagli per aiutarti a comprendere il formato ARN per diversi tipi di IAM e AWS STS risorse.

- Un utente IAM nell'account:

> **Note**
>
> Ogni nome utente IAM è univoco. Il nome utente non fa distinzione tra maiuscole e minuscole per l'utente, ad esempio durante il processo di accesso, ma la fa quando lo si utilizza in una politica o come parte di un ARN.

```
arn:aws:iam::123456789012:user/JohnDoe
```

- Altro utente con un percorso che riflette un organigramma:

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/JaneDoe
```

- Un gruppo di utenti IAM:

```
arn:aws:iam::123456789012:group/Developers
```

- Un gruppo di utenti IAM con un percorso:

```
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
```

- Ruolo IAM:

```
arn:aws:iam::123456789012:role/S3Access
```

- Un ruolo collegato al servizio:

```
arn:aws:iam::123456789012:role/aws-service-role/access-analyzer.amazonaws.com/
AWSServiceRoleForAccessAnalyzer
```

- Un ruolo di servizio:

```
arn:aws:iam::123456789012:role/service-role/QuickSightAction
```

- Policy gestita:

```
arn:aws:iam::123456789012:policy/ManageCredentialsPermissions
```

- Profilo dell'istanza che può essere associato a un'istanza Amazon EC2:

```
arn:aws:iam::123456789012:instance-profile/Webserver
```

- Un utente federato identificato in IAM come "Paulo":

```
arn:aws:sts::123456789012:federated-user/Paulo
```

- Sessione attiva di qualcuno che ha assunto il "Ruolo-contabilità" con il nome di sessione di ruolo di sessione "Mary":

```
arn:aws:sts::123456789012:assumed-role/Accounting-Role/Mary
```

- Rappresenta la sessione del chiamante quando viene utilizzata come risorsa in una chiamata API, ad esempio l' AWS STS SetContextAPI, che opera sulla sessione chiamante:

```
arn:aws:sts::123456789012:self
```

- Dispositivo di autenticazione a più fattori assegnato all'utente denominato Jorge:

```
arn:aws:iam::123456789012:mfa/Jorge
```

- Certificato del server:

```
arn:aws:iam::123456789012:server-certificate/ProdServerCert
```

- Certificato server con un percorso che riflette un organigramma:

```
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/
ProdServerCert
```

- Provider di identità (SAML e OIDC):

```
arn:aws:iam::123456789012:saml-provider/ADFSProvider
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
arn:aws:iam::123456789012:oidc-provider/server.example.org
```

- Provider di identità OIDC con un percorso che riflette l'URL di un provider di identità OIDC di Amazon EKS:

```
arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-west-2.amazonaws.com/id/
a1b2c3d4567890abcdefEXAMPLE11111
```

Un altro importante ARN è l'ARN dell'utente root. Sebbene questa non sia una risorsa IAM, bisognerebbe essere a conoscenza del formato di questo ARN. Viene spesso utilizzato nell'elemento `Principal` di una policy basata su risorse.

- Account AWS Visualizza quanto segue:

```
arn:aws:iam::123456789012:root
```

L'esempio seguente mostra una policy che può essere assegnata a Richard per la gestione autonoma delle sue chiavi di accesso. La risorsa è l'utente IAM Richard.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ManageRichardAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:GetUser"
            ],
            "Resource": "arn:aws:iam::*:user/division_abc/subdivision_xyz/Richard"
        },
        {
            "Sid": "ListForConsole",
            "Effect": "Allow",
            "Action": "iam:ListUsers",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Quando utilizzi gli ARN per identificare le risorse in una policy IAM, puoi includere le variabili delle policy. Le variabili dei criteri possono includere segnaposti per le informazioni di runtime (ad esempio il nome dell'utente) come parte dell'ARN. Per ulteriori informazioni, consulta la sezione Elementi delle policy IAM: variabili e tag

Utilizzo di caratteri jolly e percorsi negli ARN

Puoi utilizzare i caratteri jolly nella parte *risorsa* dell'ARN per specificare più utenti, gruppi di utenti o policy. Ad esempio, per specificare tutti gli utenti che lavorano su un prodotto denominato product_1234, puoi utilizzare:

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/*
```

Se hai utenti i cui nomi iniziano con la stringa `app_`, puoi fare riferimento a tutti quelli con il seguente ARN.

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/app_*
```

Per specificare tutti gli utenti, i gruppi di utenti o le politiche del tuo ARN Account AWS, usa un carattere jolly dopo `user/group/`, o `policy/` parte dell'ARN, rispettivamente.

```
arn:aws:iam::123456789012:user/*
arn:aws:iam::123456789012:group/*
arn:aws:iam::123456789012:policy/*
```

Se si specifica il seguente ARN per un utente `arn:aws:iam::111122223333:user/*`, questo corrisponderà a entrambi gli esempi seguenti.

```
arn:aws:iam::111122223333:user/JohnDoe
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/JaneDoe
```

Ma, se specifichi il seguente ARN per un utente `arn:aws:iam::111122223333:user/division_abc*`, corrisponderà al secondo esempio, ma non al primo.

```
arn:aws:iam::111122223333:user/JohnDoe
```

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/JaneDoe
```

Non utilizzare caratteri jolly nella parte `user/`, `group/`o `policy/` dell'ARN. Ad esempio, IAM non consente quanto segue:

```
arn:aws:iam::123456789012:u*
```

Example Esempio di utilizzo di percorsi e ARN per un gruppo di utenti basato su progetto

I percorsi non possono essere creati o modificati nella AWS Management Console. Per utilizzare i percorsi è necessario utilizzare la risorsa utilizzando l' AWS API AWS CLI, o gli strumenti per Windows. PowerShell

In questo esempio, Jules nel gruppo di utenti Marketing_Admin crea un gruppo basato su progetto all'interno del percorso /marketing/. Jules assegna gli utenti provenienti da diverse parti dell'azienda al gruppo di utenti. Questo esempio illustra come il percorso di un utente non sia correlato ai gruppi di utenti in cui si trova l'utente.

Il gruppo marketing sta per lanciare un nuovo prodotto, quindi Jules crea un nuovo gruppo nel percorso /marketing/, chiamandolo Widget_Launch. Jules assegna al gruppo la seguente policy, che fornisce l'accesso al gruppo di utenti a oggetti nella parte del `example_bucket` progettato appositamente per il lancio del prodotto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::example_bucket/marketing/newproductlaunch/widget/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket*",
      "Resource": "arn:aws:s3:::example_bucket",
      "Condition": {"StringLike": {"s3:prefix": "marketing/newproductlaunch/widget/*"}}
    }
  ]
}
```

Jules assegna quindi al gruppo di utenti gli utenti coinvolti nel lancio del prodotto. Fra questi ci sono Patricia ed Eli dal percorso /marketing/. Sono inclusi anche Chris e Chloe dal percorso /vendite/ e Aline e Jim dal percorso /legale/.

Identificatori univoci

Quando IAM crea un utente, un gruppo di utenti, un ruolo, una policy, un profilo dell'istanza o un certificato server, assegna a ciascuna risorsa un ID univoco. L'ID univoco ha il seguente aspetto:

```
AIDAJQABLZS4A3QDU576Q
```

In linea di massima, vengono utilizzati nomi descrittivi e ARN quando si lavora con le risorse IAM. In questo modo non è necessario conoscere l'ID univoco per una risorsa specifica. Tuttavia, a volte l'ID univoco può risultare utile, se l'utilizzo di nomi descrittivi non risulta pratico.

Un esempio riutilizza nomi descrittivi nel tuo Account AWS. All'interno dell'account, il nome descrittivo di un utente, di un gruppo di utenti, di un ruolo o di una policy deve essere univoco. Ad esempio, potresti creare un utente IAM denominato `John`. La tua azienda utilizza Amazon S3 e dispone di un bucket con cartelle per ogni dipendente. L'utente IAM `John` è un membro di un gruppo di utenti IAM denominato `User-S3-Access` con autorizzazioni che consentono agli utenti di accedere solo alle loro cartelle nel bucket. Per un esempio di come creare una policy basata sull'identità che consenta agli utenti IAM di accedere al loro oggetto del bucket in S3 utilizzando il nome descrittivo degli utenti, consulta Amazon S3: consente agli utenti IAM di accedere alla propria directory home S3, in modo programmatico e nella console.

Supponiamo che il dipendente denominato David lasci l'azienda e che il suo utente IAM denominato `John` venga eliminato. Successivamente, viene assunto un altro dipendente con lo stesso nome e viene creato un nuovo utente IAM, anch'esso denominato `John`. Aggiungi il nuovo utente IAM denominato `John` al gruppo di utenti IAM esistente `User-S3-Access`. Se la policy associata al gruppo di utente specifica il nome descrittivo dell'utente IAM `John`, la policy consente al nuovo John di accedere alle informazioni lasciate dal John precedente.

Come regola generale, ti consigliamo di specificare la ARN per la risorsa nelle policy invece del suo ID univoco. Tuttavia, ogni utente IAM dispone di un ID univoco, anche se crei un nuovo utente IAM che riutilizza un nome descrittivo che avevi eliminato in precedenza. In questo esempio, il vecchio utente IAM `John` e il nuovo utente IAM `John` hanno ID univoci differenti. È possibile creare policy basate sulle risorse che concedono l'accesso in base all'ID univoco e non solo per nome utente. In questo modo si riduce la possibilità che si possa inavvertitamente concedere l'accesso alle informazioni che un dipendente non dovrebbe avere.

L'esempio seguente mostra come specificare ID univoci nell'elemento `Principal` di una policy basata su risorse.

```
"Principal": {
  "AWS": [
    "arn:aws:iam::111122223333:role/role-name",
    "AIDACKCEVSQ6C2EXAMPLE",
    "AROADBQP57FF2AEXAMPLE"
  }
```

L'esempio seguente mostra come specificare ID univoci nell'elemento `Condition` di una policy che utilizza la chiave di condizione globale `aws:userid`.

```
"Condition": {
    "StringLike": {
      "aws:userId": [
        "AIDACKCEVSQ6C2EXAMPLE",
        "AROADBQP57FF2AEXAMPLE:role-session-name",
        "AROA1234567890EXAMPLE:*",
        "111122223333"
      ]
    }
  }
```

Un altro esempio è dato dagli ID utente che possono tornare utili quando ti trovi a gestire un database personale (o un altro archivio) con informazioni sul ruolo o sugli utenti IAM. L'ID univoco può fornire un identificatore univoco per ogni ruolo o utente IAM che viene creato. Questo è il caso di ruoli o utenti IAM che riutilizzano un nome, come nell'esempio precedente.

Approfondimento dei prefissi ID univoci

IAM usa i seguenti prefissi per indicare il tipo di risorsa a cui si applica ciascun ID univoco. I prefissi possono variare in base a quando sono stati creati.

Prefix	Tipo di risorsa
ABIA	AWS STS token service bearer
ACCA	Credenziali specifiche per contesto

Prefix	Tipo di risorsa
AGPA	Gruppo di utenti
AIDA	Utente IAM
AIPA	Profilo dell'istanza Amazon EC2
AKIA	Chiave di accesso
ANPA	Policy gestita
ANVA	Versione in una policy gestita
APKA	Chiavi pubbliche
AROA	Ruolo
ASCA	Certificato
ASIA	Gli ID chiave di accesso temporanea (AWS STS) utilizzano questo prefisso, ma sono univoci solo in combinazione con la chiave di accesso segreta e il token di sessione.

Ottenere l'identificatore univoco

L'ID univoco per una risorsa IAM non è disponibile nella console IAM. Per ottenere l'ID univoco, puoi utilizzare i seguenti AWS CLI comandi o chiamate API IAM.

AWS CLI:

- get-caller-identity
- get-group
- get-role
- get-user
- get-policy
- get-instance-profile

- get-server-certificate

API IAM:

- GetCallerIdentity
- GetGroup
- GetRole
- GetUser
- GetPolicy
- GetInstanceProfile
- GetServerCertificate

IAM e AWS STS quote

AWS Identity and Access Management (IAM) e AWS Security Token Service (STS) dispongono di quote che limitano la dimensione degli oggetti. Questi servizi limitano anche la modalità di denominazione di un oggetto, il numero di oggetti che è possibile creare e il numero di caratteri che è possibile utilizzare quando si passa un oggetto.

> **Note**
>
> Per ottenere informazioni a livello di account sull'utilizzo e sulle quote di IAM, utilizza l'operazione GetAccountSummaryAPI o il comando. get-account-summary AWS CLI

Requisiti del nome IAM

I nomi IAM sono caratterizzati dalle limitazioni e dai requisiti seguenti:

- I documenti delle policy possono contenere solo i seguenti caratteri Unicode: tabulatore orizzontale (U+0009), avanzamento riga (U+000A), ritorno a capo (U+000D) e i caratteri compresi fra U+0020 a U+00FF.
- I nomi di utenti, gruppi, ruoli, policy, profili dell'istanza, certificati server e percorsi devono essere alfanumerici, inclusi i seguenti caratteri comuni: più (+), uguale (=), virgola (,), punto (.), a (@), trattino basso (_) e trattino (-). I nomi dei percorsi devono iniziare e terminare con una barra (/).

- I nomi di utenti, gruppi, ruoli e profili di istanze devono essere univoci all'interno dell'account. Non viene applicata la distinzione fra maiuscole e minuscole. Ad esempio, non è possibile creare un gruppo **`ADMINS`** e un altro **`admins`**.
- Il valore dell'ID esterno che una terza parte utilizza per assumere un ruolo deve avere un minimo di 2 caratteri e un massimo di 1.224 caratteri. Il valore deve essere alfanumerico senza spazi. Può anche includere i seguenti simboli: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), due punti (:), barra (/) e trattino (-). Per ulteriori informazioni sull'ID esterno, consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.
- I nomi delle policy in linea devono essere univoci per l'utente, gruppo o ruolo in cui sono incorporati. I nomi possono contenere qualsiasi carattere latino di base (ASCII), ad eccezione di alcuni caratteri riservati: barra rovesciata (\), barra (/), asterisco (*), punto interrogativo (?) e spazio. Questi caratteri sono riservati in base a RFC 3986, sezione 2.2.
- Le password utente (profili di accesso) possono contenere tutti i caratteri latini di base (ASCII).
- Account AWS Gli alias ID devono essere univoci per tutti AWS i prodotti e devono essere alfanumerici secondo le convenzioni di denominazione DNS. Un alias deve essere in lettere minuscole, non può iniziare o terminare con un trattino, non può contenere due trattini consecutivi e non può essere un numero di 12 cifre.

Per un elenco dei caratteri latini di base (ASCII), consulta la Library of Congress Basic Latin (ASCII) Code Table.

IAM Quote oggetto

Le quote, note anche come limiti in, sono i valori massimi per le risorse AWS, le azioni e gli elementi presenti in. Account AWSÈ possibile utilizzare Service Quotas per gestire le quote IAM.

Per l'elenco degli endpoint e delle quote dei servizi IAM, consulta Endpoint e quote di AWS Identity and Access Management nella Riferimenti generali di AWS

Richiesta di un aumento delle quote

1. Segui la procedura di accesso appropriata per il tuo tipo di utente, come descritto in Come accedere ad AWS nella Guida per l'utente di AWS Sign-In per accedere alla AWS Management Console.
2. Apri la console Service Quotas.
3. Nel pannello di navigazione, scegli Servizi AWS (servizi AWS).

4. Sulla barra di navigazione, selezionare la regione US East (N. Virginia). Quindi cercare **`IAM`**.
5. Scegli AWS Identity and Access Management (IAM), seleziona una quota e segui le istruzioni per richiedere un aumento di quota.

Per ulteriori informazioni, consulta Richiesta di un aumento di quota nel Guida per l'utente di Service Quotas.

Per un esempio di come richiedere un aumento della quota IAM utilizzando la console Service Quotas, guarda il video seguente.

Richiedi un aumento della quota IAM utilizzando la console Service Quotas.

Puoi richiedere un aumento delle quote predefinite per le quote IAM regolabili. Le richieste fino a maximum quota vengono automaticamente approvate e completate in pochi minuti.

Nella tabella seguente sono elencate le risorse per le quali è possibile approvare automaticamente gli aumenti di quota.

Quote modificabili per le risorse IAM

Risorsa	Quota predefinita	Quota massima
Policy gestite dal cliente per account	1500	5000
Gruppi per account	300	500
Profili di istanza per account	1000	5000
Policy gestite per ruolo	10	20
Policy gestite per utente	10	20
Lunghezza della policy di attendibilità del ruolo	2048 caratteri	4.096 caratteri
Ruoli per account	1000	5000
Certificati server per account	20	1000

Quote Sistema di analisi degli accessi AWS IAM

Per l'elenco degli endpoint e delle quote dei servizi Sistema di analisi degli accessi AWS IAM, consulta Endpoint e quote di Sistema di analisi degli accessi AWS IAM nella Riferimenti generali di AWS.

Quote di IAM Roles Anywhere

Per l'elenco degli endpoint e delle quote dei servizi IAM Roles Anywhere, consulta Endpoint e quote di AWS Identity and Access Management Roles Anywhere nella Riferimenti generali di AWS.

Limiti di caratteri di IAM e STS

Di seguito sono riportati i numeri massimi di caratteri e i limiti di dimensione per IAM e AWS STS. Non puoi richiedere un aumento per i seguenti limiti.

Descrizione	Limite
Alias per un ID Account AWS	3-63 caratteri
Per policy inline	Non esiste un limite per le policy inline che puoi aggiungere a un utente, a un gruppo o a un ruolo IAM. Tuttavia, la dimensione cumulativa della policy (ovvero, la somma delle dimension i di tutte le policy in linea) per ciascuna entità non può superare i seguenti limiti: • La dimensione della policy dell'utente non può superare i 2.048 caratteri. • La dimensione della policy del ruolo non può superare i 10.240 caratteri. • La dimensione della policy del gruppo non può superare i 5.120 caratteri.

Descrizione	Limite
	Note IAM non conta gli spazi nel calcolo per determinare le dimensioni di una policy rispetto a tali limiti.
Per policy gestite	• La dimensione di ciascuna policy gestita non può superare i 6.144 caratteri. **Note** IAM non conta gli spazi nel calcolo per determinare le dimensioni di una policy rispetto a tale limite.
Group name (Nome gruppo)	128 caratteri
Nome del profilo dell'istanza	128 caratteri
Password per un profilo di accesso	1-128 caratteri
Path	512 caratteri
Policy name (Nome policy)	128 caratteri
Role Name (Nome ruolo)	64 caratteri **Important** Se intendi utilizzare un ruolo con la funzione Cambia ruolo in AWS Management Console, la combinazione `Path` non `RoleName` può superare i 64 caratteri.

Descrizione	Limite
Durata della sessione del ruolo	12 ore Quando assumi un ruolo dall'API AWS CLI o, puoi utilizzare il parametro `duration-seconds` CLI o il parametro `DurationSeconds` API per richiedere una sessione di ruolo più lunga. È possibile specificare un valore compreso fra 900 secondi (15 minuti) fino all'impostazione massima della durata consentita per il ruolo, che può variare da 1 a 12 ore. Se non specifichi un valore per il parametro `DurationSeconds` le tue credenziali di sicurezza rimarranno valide per un'ora. Agli utenti IAM che cambiano ruoli nella console viene concessa la durata massima della sessione o il tempo rimanente nella sessione dell'utente, a seconda di quale sia minore. L'impostazione di durata massima delle sessioni non limita le sessioni assunte dai servizi AWS . Per informazioni su come visualizzare il valore massimo per il ruolo, consulta Visualizzazione dell'impostazione di durata massima della sessione per un ruolo.
Nome della sessione del ruolo	64 caratteri

Descrizione	Limite
Policy di sessione del ruolo	• La dimensione del documento di policy JSON inviato e tutti i caratteri ARN delle policy gestite inviate non possono superare i 2.048 caratteri. • È possibile passare un massimo di 10 ARN delle policy gestite quando si crea una sessione. • Al momento della creazione a livello di programmazione di una sessione temporane a per un ruolo o un utente federato è possibile passare un solo documento JSON di policy. • Inoltre, una AWS conversione comprime le politiche di sessione e i tag di sessione passati in un formato binario compresso con un limite separato. L'elemento della risposta `PackedPolicySize` indica in percentuale la consistenza di policy e tag della richiesta rispetto al limite di dimensione superiore. • Ti consigliamo di passare i criteri di sessione utilizzando l' AWS API AWS CLI or. AWS Management Console Potrebbero aggiunger e ulteriori informazioni sulla sessione della console alla politica compressa.

Descrizione	Limite
Tag di sessione per il ruolo	• I tag di sessione devono soddisfare il limite della chiave del tag di 128 caratteri e il limite del valore del tag di 256 caratteri. • È possibile passare fino a 50 tag di sessione. • Una AWS conversione comprime i criteri di sessione e i tag di sessione passati in un formato binario compresso con un limite separato. È possibile passare i tag di sessione utilizzando l' AWS API AWS CLI or. L'elemento della risposta `PackedPolicySize` indica in percentuale la consistenza di policy e tag della richiesta rispetto al limite di dimensione superiore.
Risposta di autenticazione SAML codificata con base64	100.000 caratteri Questo limite di caratteri si applica all'operazione della CLI `assume-role-with-saml` o dell'API `AssumeRoleWithSAML`.
Chiave tag	128 caratteri Questo limite di caratteri si applica ai tag sulle risorse IAM e ai tag di sessione.
Valore tag	256 caratteri Questo limite di caratteri si applica ai tag sulle risorse IAM e ai tag di sessione. I valori dei tag possono essere vuoti, ciò significa che possono avere una lunghezza di 0 caratteri.

Descrizione	Limite
ID univoci creati da IAM	128 caratteri Ad esempio: • ID utente che iniziano con `AIDA` • ID di gruppi che iniziano con `AGPA` • ID di ruoli che iniziano con `AROA` • ID di policy gestite che iniziano con `ANPA` • ID di certificati server che iniziano con `ASCA` **Note** Questo elenco non è completo e non fornisce alcuna garanzia che gli ID di un determinato tipo inizino sempre con la combinazione di lettere specificata.
Nome utente	64 caratteri

Endpoint VPC di interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e (). AWS Security Token Service AWS STS Puoi utilizzare questa connessione AWS STS per consentire di comunicare con le tue risorse nel tuo VPC senza passare attraverso la rete Internet pubblica.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per connettere il tuo VPC AWS STS, definisci un'interfaccia VPC endpoint per. AWS STS L'endpoint fornisce una connettività affidabile e scalabile AWS STS senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta Che cos'è Amazon VPC? nella Guida per l'utente di Amazon VPC

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, vedere Servizi AWS PrivateLink . AWS

Le seguenti informazioni sono destinate agli utenti di Amazon VPC. Per ulteriori informazioni, consulta Nozioni di base su Amazon VPC nella Guida per l'utente di Amazon VPC.

Disponibilità

AWS STS attualmente supporta gli endpoint VPC nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)

- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Europa (Stoccolma)
- Europa (Zurigo)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Crea un endpoint VPC per AWS STS

Per iniziare a utilizzarlo AWS STS con il tuo VPC, crea un endpoint VPC di interfaccia per. AWS STS Per ulteriori informazioni, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Amazon VPC User Guide.

Dopo aver creato l'endpoint VPC, devi utilizzare l'endpoint regionale corrispondente per inviare le tue richieste. AWS STS AWS STS consiglia di utilizzare entrambi i `setEndpoint` metodi `setRegion` e per effettuare chiamate a un endpoint regionale. Puoi utilizzare il metodo `setRegion` da solo per regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). In questo caso, le chiamate sono indirizzate all'endpoint regionale STS. Per ulteriori informazioni su come abilitare manualmente una regione, consulta Gestione delle regioni AWS in Riferimenti generali di AWS. Se utilizzi il metodo `setRegion` da solo per regioni abilitate per default, le chiamate vengono indirizzate all'endpoint globale di `https://sts.amazonaws.com` .

Quando utilizzi endpoint regionali, AWS STS chiama altri AWS servizi utilizzando endpoint pubblici o endpoint VPC con interfaccia privata, a seconda di quale siano in uso. Ad esempio, supponi di aver creato un endpoint VPC di interfaccia AWS STS e di aver già richiesto credenziali temporanee AWS STS alle risorse che si trovano nel tuo VPC. In tal caso, queste credenziali iniziano a fluire attraverso l'endpoint VPC dell'interfaccia per impostazione predefinita. Per ulteriori informazioni sull'utilizzo di richieste regionali, consulta. AWS STSGestire AWS STS in un Regione AWS

AWS servizi che funzionano con IAM

I AWS servizi elencati di seguito sono raggruppati in ordine alfabetico e includono informazioni sulle funzionalità IAM che supportano:

- Servizio: puoi scegliere il nome di un servizio per visualizzare la AWS documentazione sull'autorizzazione e l'accesso IAM a quel servizio.
- Operazioni: è possibile specificare singole operazioni in una policy. Se il servizio non supporta questa funzionalità, vengono selezionate Tutte le operazioni nell'editor visivo. In un documento di policy JSON, è necessario utilizzare `*` nell'elemento `Action`. Per un elenco delle azioni in ogni servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi.
- Autorizzazioni a livello di risorsa: è possibile utilizzare gli ARN per specificare singole risorse nella policy. Se il servizio non supporta questa funzionalità, viene selezionata l'opzione Tutte le risorse nell'editor visivo della policy. In un documento di policy JSON, è necessario utilizzare `*` nell'elemento `Resource`. Alcune operazioni, ad esempio le operazioni `List*`, non supportano la specifica di un ARN perché sono progettate per restituire più risorse. Se un servizio supporta questa funzionalità per alcune risorse ma non per altre, il limite viene indicato con la dicitura Partial (Parziale) nella tabella. Per ulteriori informazioni, consulta la documentazione per quel servizio.
- Policy basate su risorse: è possibile collegare policy basate su risorse a una risorsa all'interno del servizio. Le policy basate su risorse includono un elemento `Principal` che consente di specificare quali identità IAM possono accedere a tale risorsa. Per ulteriori informazioni, consulta Policy basate sulle identità e policy basate su risorse.
- ABAC (autorizzazione basata su tag): per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento di condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`. Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Partial (Parziale). Per ulteriori informazioni sulla definizione delle autorizzazioni basate su attributi quali i tag, consulta A cosa serve ABAC? AWS. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC).
- Credenziali temporanee: puoi utilizzare credenziali a breve termine ottenute quando accedi tramite IAM Identity Center, cambi ruolo nella console o generate utilizzando l' AWS STS API AWS CLI o AWS . È possibile accedere ai servizi con un valore No solo utilizzando le credenziali utente IAM a lungo termine. Questo include un nome utente e una password o le chiavi di accesso utente. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM.

- Ruoli collegati al servizio: un ruolo collegato ai servizi è un tipo speciale di ruolo di servizio che consente al servizio di accedere alle risorse di altri servizi per conto dell'utente. Selezionare il collegamento Sì o Parziale per visualizzare la documentazione per i servizi che supportano questi ruoli. Questa colonna non indica se il servizio utilizza ruoli di servizio standard. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi.
- Ulteriori informazioni: se un servizio non supporta completamente una funzionalità, è possibile esaminare le note a piè di pagina per visualizzare le limitazioni e i collegamenti alle informazioni correlate.

Servizi supportati da IAM

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Account Management	 Sì	 Sì	 No	 No	 Sì	 No
AWS Activate Console	 Sì	 No	 No	 No	 Sì	 No
AWS Amplify Amministratore	 Sì	 Sì	 No	 No	 Sì	 No
AWS Amplify	 Sì	 Sì	 No	 Parziale	 Sì	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Amplify Generatore di interfacce utente	Sì	Sì	No	Sì	Sì	No
API di Apache Kafka per i cluster di Amazon MSK	Sì	Sì	No	No	Sì	No
Gateway Amazon API	Sì	Sì	Sì	No	Sì	Sì
Gestione di Gateway Amazon API	Sì	Sì	No	Sì	Sì	No
Gestione di Gateway Amazon API V2	Sì	Sì	No	Sì	Sì	No
AWS App2Container	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS AppConfig	Sì	Sì	No	Sì	Sì	No
AWS AppFabric	Sì	Sì	No	Sì	Sì	No
Amazon AppFlow	Sì	Sì	No	Sì	Sì	No
Amazon AppIntegrations	Sì	Sì	No	Sì	Sì	Sì
Application Auto Scaling	Sì	Sì	No	Sì	Sì	Sì
AWS Application Cost Profiler	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Applicazione Discovery Arsenal	Sì	No	No	No	Sì	No
AWS Application Discovery Service	Sì	No	No	No	Sì	Sì
AWS Application Migration Service	Sì	Sì	No	Sì	Sì	Sì
AWS Servizio di trasformazione delle applicazioni	Sì	No	No	No	Sì	No
AWS App Mesh	Sì	Sì	No	Sì	Sì	Sì
AWS App Mesh Anteprima	Sì	Sì	No	No	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS App Runner	Sì	Sì	No	Sì	Sì	Sì
Amazon AppStream 2.0	Sì	Sì	No	Sì	Sì	No
AWS AppSync	Sì	Sì	No	Sì	Sì	No
AWS Artifact	Sì	Sì	No	No	Sì	No
Amazon Athena	Sì	Sì	No	Sì	Sì	No
AWS Audit Manager	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Auto Scaling	Sì	No	No	No	Sì	Sì
AWS Scambio di dati B2B	Sì	Sì	No	Sì	Sì	No
AWS Backup	Sì	Sì	Sì	Sì	Sì	Sì
AWS Backup Gateway	Sì	Sì	No	Sì	Sì	No
AWS Backup archiviazione	Sì	No	No	No	Sì	No
AWS Batch	Sì	Parziale	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Bedrock	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS Billing and Cost Management	 Sì	 No	 No	 No	 Sì	 Sì
AWS Billing and Cost Management Esportazioni di dati	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS Billing Conductor	 Sì	 Sì	 No	 Sì	 Sì	 No
Amazon Braket	 Sì	 Sì	 No	 Sì	 Sì	 Sì
AWS Servizio di bilancio	 Sì	 Sì	 No	 No	 No	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS BugBust	Sì	Sì	No	Sì	Sì	Sì
AWS Certificate Manager (ACM)	Sì	Sì	No	Sì	Sì	Sì
AWS Chatbot	Sì	Sì	No	No	Sì	Sì
Amazon Chime	Sì	Sì	No	Sì	Sì	Sì
AWS Clean Rooms	Sì	Sì	No	Sì	Sì	No
AWS Clean Rooms ML	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Client VPN	Sì	Sì	No	No	Sì	Sì
AWS Cloud9	Sì	Sì	Sì	Sì	Sì	Sì
Cloud AWS API di controllo	Sì	No	No	No	Sì	No
Directory del cloud Amazon	Sì	Sì	No	No	Sì	No
AWS CloudFormation	Sì	Sì	No	Sì	Sì	No
Amazon CloudFront	Sì	Sì	No	Parziale	Sì	Parziale(Informazioni)

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon CloudFront KeyValueStore	Sì	Sì	No	No	Sì	No
AWS CloudHSM	Sì	Sì	No	Sì	Sì	Sì
AWS Cloud Map	Sì	Sì	No	Sì	Sì	No
Amazon CloudSearch	Sì	Sì	No	No	Sì	No
AWS CloudShell	Sì	Sì	No	No	Sì	No
AWS CloudTrail	Sì	Sì	Parziale (Informazioni)	Parziale (Informazioni)	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS CloudTrail Dati	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch	Sì	Sì	No	Sì	Sì	Parziale(Informazioni)
Informazioni approfondite sulle CloudWatch applicazioni Amazon	Sì	No	No	No	Sì	No
Segnali CloudWatch applicativi Amazon	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch evidentemente	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch Internet Monitor	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
CloudWatch Registri Amazon	Sì	Sì	Sì	Parziale	Sì	Sì
Monitoraggio CloudWatch di rete Amazon	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch Observability Access Manager	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch RUM	Sì	Sì	No	Sì	Sì	No
Amazon CloudWatch Synthetics	Sì	Sì	No	Sì	Sì	No
AWS CodeArtifact	Sì	Sì	Sì	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS CodeBuild	Sì	Sì	Sì (Informazioni)	Parziale (Informazioni)	Sì	No
Amazon CodeCatalyst	Sì	Sì	No	Sì	Sì	Sì
AWS CodeCommit	Sì	Sì	No	Sì	Sì	No
AWS CodeConnections	Sì	Sì	No	Sì	Sì	No
AWS CodeDeploy	Sì	Sì	No	Sì	Sì	No
AWS CodeDeploy servizio di comandi host sicuri	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon CodeGuru Profiler	Sì	Sì	No	Sì	Sì	Sì
CodeGuru Revisore Amazon	Sì	Sì	No	Sì	Sì	Sì
CodeGuru Sicurezza Amazon	Sì	Sì	No	Sì	Sì	No
AWS CodePipeline	Sì	Parziale	No	Sì	Sì	No
AWS CodeStar	Sì	Parziale	No	Sì	Sì	No
AWS CodeStar Connessioni	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Notifiche AWS CodeStar	Sì	Sì	No	Sì	Sì	Sì
Amazon CodeWhisperer	Sì	Sì	No	Sì	Sì	Sì
Amazon Cognito	Sì	Sì	No	Sì	Sì	Sì
Amazon Cognito Sync	Sì	Sì	No	No	Sì	Sì
Pool di utenti Amazon Cognito	Sì	Sì	No	Sì	Sì	Sì
Amazon Comprehend	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Comprehend Medical	Sì	No	No	No	Sì	No
AWS Compute Optimizer	Sì	No	No	No	Sì	Sì
AWS Config	Sì	Parziale (Informazioni)	No	Sì	Sì	Sì
Amazon Connect	Sì	Sì	No	Sì	Sì	Sì
Casi di Amazon Connect	Sì	Sì	No	Sì	Sì	No
Profili cliente Amazon Connect	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Comunicazioni Amazon Connect in uscita con volumi elevati	Sì	Sì	No	Sì	Sì	No
Amazon Connect Voice ID	Sì	Sì	No	Sì	Sì	No
AWS Console Mobile Application	Sì	Sì	No	No	Sì	No
AWS Fatturazione consolidata	Sì	No	No	No	Sì	No
AWS Catalogo di controllo	Sì	Sì	No	No	Sì	No
AWS Control Tower	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Cost and Usage Report	Sì	Sì	No	No	Sì	No
AWS Cost Explorer	Sì	Sì	No	Sì	Sì	No
Centrale ottimizzazione costi AWS	Sì	No	No	No	Sì	No
Servizio di verifica clienti AWS	Sì	No	No	No	Sì	No
AWS Database Migration Service	Sì	Sì	No (Informazioni)	Sì	Sì	Sì
Database Query Metadata Service	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Data Exchange	Sì	Sì	No	Sì	Sì	No
Amazon Data Lifecycle Manager	Sì	Sì	No	Sì	Sì	No
AWS Data Pipeline	Sì	Sì	No	Parziale	Sì	No
AWS DataSync	Sì	Sì	No	Sì	Sì	Sì
Amazon DataZone	Sì	No	No	No	Sì	No
AWS Deadline Cloud	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS DeepComposer	Sì	Sì	No	Sì	Sì	No
AWS DeepRacer	Sì	Sì	No	Sì	Sì	Sì
Amazon Detective	Sì	Sì	No	Sì	Sì	No
AWS Device Farm	Sì	Sì	No	Sì	Sì	Sì
Amazon DevOps Guru	Sì	Sì	No	No	Sì	Sì
Strumenti di diagnostica AWS	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Direct Connect	Sì	Sì	No	Sì	Sì	Sì
AWS Directory Service	Sì	Sì	No	Sì	Sì	No
Cluster elastici Amazon DocumentDB	Sì	Sì	No	Sì	Sì	Sì
Acceleratore Amazon DynamoDB (DAX)	Sì	Sì	No	No	Sì	Sì
Amazon DynamoDB	Sì	Sì	Sì	No	Sì	No
Amazon Elastic Compute Cloud (Amazon EC2)	Sì	Parziale	No	Sì	Sì	Parziale (Informazioni)

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Dimensionamento automatico Amazon EC2	Sì	Sì	No	Sì	Sì	Sì
EC2 Image Builder	Sì	Sì	No	Sì	Sì	Sì
Amazon EC2 Instance Connect	Sì	Sì	No	No	Sì	Sì
Amazon ElastiCache	Sì	Sì	No	Sì	Sì	Sì
AWS Elastic Beanstalk	Sì	Parziale	No	Sì	Sì	Sì
Amazon Elastic Block Store (Amazon EBS)	Sì	Parziale	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Elastic Container Registry (Amazon ECR)	Sì	Sì	Sì	Sì	Sì	Sì
Amazon Elastic Container Registry Pubblico (Amazon ECR pubblico)	Sì	Sì	No	Sì	Sì	No
Amazon Elastic Container Service (Amazon ECS)	Sì	Parzia (Informazioni)	No	Sì	Sì	Sì
AWS Elastic Disaster Recovery	Sì	Sì	No	Sì	Sì	Sì
Amazon Elastic File System (Amazon EFS)	Sì	Sì	Sì	Parziale	Sì	Sì
Amazon Elastic Inference	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Elastic Kubernetes Service (Amazon EKS)	Sì	Sì	No	Sì	Sì	Sì
Auth Amazon Elastic Kubernetes Service (Amazon EKS)	Sì	Sì	No	No	Sì	No
AWS Elastic Load Balancing	Sì	Parziale	No	Parziale	Sì	Sì
Amazon Elastic Transcoder	Sì	Sì	No	No	Sì	No
AWS Servizio di attivazione di elettrodomestici e software elementari	Sì	Sì	No	Sì	Sì	No
AWS Apparecchiature e software elementali	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Elemental MediaConnect	Sì	Sì	No	No	Sì	Sì
AWS Elemental MediaConvert	Sì	Sì	No	Sì	Sì	No
AWS Elemental MediaLive	Sì	Sì	No	Sì	Sì	No
AWS Elemental MediaPackage	Sì	Sì	No	Sì	Sì	Parziale(Informazioni)
AWS Elemental MediaPackage V2	Sì	Sì	No	Sì	Sì	No
AWS Elemental MediaPackage VOD	Sì	Sì	No	Sì	Sì	Parziale(Informazioni)

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Elemental MediaStore	Sì	Sì	Sì	Sì	Sì	No
AWS Elemental MediaTailor	Sì	Sì	No	Sì	Sì	Sì
AWS Casi Elemental Support	Sì	No	No	No	Sì	No
AWS Contenuto di Elemental Support	Sì	No	No	No	Sì	No
Amazon EMR	Sì	Sì	No	Sì	Sì	Sì
Amazon EMR su EKS	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon EMR Serverless	Sì	Sì	No	Sì	Sì	Sì
AWS Risoluzione dell'entità	Sì	Sì	No	Sì	Sì	No
Amazon EventBridge	Sì	Sì	Sì	Sì	Sì	No
EventBridge Tubi Amazon	Sì	Sì	No	Sì	Sì	No
Amazon EventBridge Scheduler	Sì	Sì	No	Sì	Sì	No
EventBridge Schemi Amazon	Sì	Sì	Sì	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Servizio di iniezione dei guasti	Sì	Sì	No	Sì	Sì	Sì
Amazon FinSpace	Sì	Sì	No	Sì	Sì	Sì
Amazon FinSpace API	Sì	Sì	No	No	Sì	No
AWS Firewall Manager	Sì	Sì	No	Sì	Sì	Parziale
Fleet Hub for AWS IoT Device Management	Sì	Sì	No	Sì	Sì	No
Amazon Forecast	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Fraud Detector	Sì	Sì	No	Sì	Sì	No
FreeRTOS	Sì	Sì	No	Sì	Sì	No
AWS Livello gratuito	Sì	No	No	No	Sì	No
Amazon FSx	Sì	Sì	No	Sì	Sì	Sì
Amazon GameLift	Sì	Sì	No	Sì	Sì	No
AWS Global Accelerator	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Glue	Sì	Sì	Sì	Parziale	Sì	No
AWS Glue DataBrew	Sì	Sì	No	Sì	Sì	No
AWS Ground Station	Sì	Sì	No	Sì	Sì	Sì
Amazon Ground Truth Labeling	Sì	No	No	No	Sì	No
Amazon GuardDuty	Sì	Sì	No	Sì	Sì	Sì
AWS Health API e notifiche	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS HealthImaging	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS HealthLake	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS HealthOmics	 Sì	 Sì	 No	 Sì	 Sì	 No
Amazon Honeycode	 Sì	 Sì	 No	 No	 Sì	 No
AWS IAM Identity Center	 Sì	 Sì	 No	 Parziale	 Sì	 Sì
Directory di IAM Identity Center	 Sì	 No	 No	 No	 Sì	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Archivio identità di IAM Identity Center	Sì	Sì	No	No	Sì	No
Servizio OIDC IAM Identity Center	Sì	Sì	No	No	Sì	No
AWS Identity and Access Management (IAM)	Sì	Sì	Parziale (Informazioni)	Parziale (Informazioni)	Parziale (Informazioni)	No
AWS Identity and Access Management Access Analyzer	Sì	Sì	No	Sì	Sì	Parziale
AWS Identity and Access Management Ruoli ovunque	Sì	Sì	No	Sì	Sì	Sì
AWS Autenticazione di Identity Store	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Sincronizzazione identità	Sì	Sì	No	No	Sì	No
AWS Import/Export	Sì	No	No	No	Sì	No
Amazon Inspector	Sì	Sì	No	Sì	Sì	Sì
Amazon Inspector Classic	Sì	No	No	No	Sì	Sì
Amazon InspectorScan	Sì	No	No	No	Sì	No
Amazon Interactive Video Service	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Interactive Video Service Chat	Sì	Sì	No	Sì	Sì	No
Fatturazione AWS	Sì	No	No	No	Sì	No
AWS IoT 1-Click	Sì	Sì	No	Sì	Sì	No
AWS IoT Analytics	Sì	Sì	No	Sì	Sì	No
AWS IoT	Sì	Sì	Parziale (Informazioni)	Sì	Sì	No
AWS IoT Core Consulente per dispositivi	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS IoT Tester per dispositivi	Sì	No	No	No	Sì	No
AWS IoT Events	Sì	Sì	No	Sì	Sì	No
AWS IoT FleetWise	Sì	Sì	No	Sì	Sì	No
AWS IoT Greengrass	Sì	Sì	No	Sì	Sì	No
AWS IoT Greengrass V2	Sì	Sì	No	Parziale	Sì	No
AWS IoT Lavori DataPlane	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS IoT RoboRunner	Sì	Sì	No	No	Sì	No
AWS IoT SiteWise	Sì	Sì	No	Sì	Sì	Sì
AWS IoT TwinMaker	Sì	Sì	No	Sì	Sì	Sì
AWS IoT Wireless	Sì	Sì	No	Sì	Sì	No
AWS IQ	Sì	Sì	No	No	Sì	Sì
AWS Autorizzazioni IQ	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Kendra	Sì	Sì	No	Sì	Sì	No
Classificazione intelligente di Amazon Kendra	Sì	Sì	No	Sì	Sì	No
AWS Key Management Service (AWS KMS)	Sì	Sì	Sì	Sì	Sì	Sì
Amazon Keyspaces (per Apache Cassandra)	Sì	Sì	No	Sì	Sì	Sì
Servizio gestito da Amazon per Apache Flink	Sì	Sì	No	Sì	Sì	No
Servizio gestito da Amazon per Apache Flink V2	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Data Firehose	 Sì	 Sì	 No	 Sì	 Sì	 No
Flusso di dati Amazon Kinesis	 Sì	 Sì	 Sì	 No	 Sì	 No
Flusso di video Amazon Kinesis	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS Lake Formation	 Sì	 No	 No	 No	 Sì	 Sì
AWS Lambda	 Sì	 Sì	 Sì	 Parziale (Informazioni)	 Sì	 Parziale (Informazioni)
AWS Launch Wizard	 Sì	 No	 No	 No	 Sì	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Lex	Sì	Sì	No	Sì	Sì	Sì
Amazon Lex V2	Sì	Sì	Sì	Sì	Sì	Sì
AWS License Manager	Sì	Sì	No	Sì	Sì	Sì
AWS License Manager Gestore di abbonamenti Linux	Sì	No	No	No	Sì	No
AWS License Manager Sottoscrizioni utente	Sì	No	No	No	Sì	Sì
Amazon Lightsail	Sì	Parziale (Informazioni)	No	Parziale (Informazioni)	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Servizio di posizione Amazon	Sì	Sì	No	Sì	Sì	No
Amazon Lookout per le apparecchiature	Sì	Sì	No	Sì	Sì	No
Amazon Lookout per le metriche	Sì	Sì	No	Sì	Sì	No
Amazon Lookout per Vision	Sì	Sì	No	Sì	Sì	No
Amazon Machine Learning	Sì	Sì	No	No	Sì	No
Amazon Macie	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Mainframe Modernization	Sì	Sì	No	Sì	Sì	Sì
AWS Mainframe Modernization Test delle applicazioni	Sì	Sì	No	Sì	Sì	No
Blockchain gestita da Amazon	Sì	Sì	No	Sì	Sì	No
Query su Blockchain gestita da Amazon	Sì	No	No	No	Sì	No
Grafana gestito da Amazon	Sì	Sì	No	Sì	Sì	Sì
Amazon Managed Service per Prometheus	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Managed Streaming per Apache Kafka (MSK)	Sì	Sì	Parziale (Informazioni)	Sì	Sì	Sì
Amazon Managed Streaming per Kafka Connect	Sì	Sì	No	No	Sì	Sì
Flussi di lavoro gestiti da Amazon per Apache Airflow	Sì	Sì	No	Sì	Sì	No
Marketplace AWS	Sì	No	No	No	Sì	Sì
Marketplace AWS Catalogo	Sì	Sì	No	Sì	Sì	No
AWS Marketplace Commerce Analytics	Sì	No	No	No	No	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Servizio di implementazione Marketplace AWS	 Sì	 Sì	 No	 Sì	 Sì	 No
Marketplace AWS Individuazione	 Sì	 No	 No	 No	 Sì	 No
AWS Marketplace Entitlement Service	 Sì	 No	 No	 No	 Sì	 No
Marketplace AWS Servizio di creazione di immagini	 Sì	 No	 No	 No	 Sì	 No
Portale di gestione Marketplace AWS	 Sì	 No	 No	 No	 Sì	 No
AWS Marketplace Metering Service	 Sì	 No	 No	 No	 Sì	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Marketplace AWS Marketplace privato	Sì	No	No	No	Sì	No
Marketplace AWS Integrazione dei sistemi di approvvigionamento	Sì	No	No	No	Sì	No
Marketplace AWS Rapporti sui venditori	Sì	Sì	No	No	Sì	No
Marketplace AWS Informazioni sui fornitori	Sì	Sì	No	Sì	Sì	No
Amazon Mechanical Turk	Sì	No	No	No	Sì	No
Amazon MediaImport	Sì	No	No	No	No	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon MemoryDB for Redis	Sì	Sì	No	Sì	Sì	Sì
Amazon Message Delivery Service	Sì	No	No	No	Sì	No
Servizio Amazon Message Gateway	Sì	No	No	No	Sì	No
AWS Microservice Extractor for .NET	Sì	No	No	No	Sì	No
AWS Crediti del Migration Acceleration Program	Sì	Sì	No	No	Sì	No
AWS Migration Hub	Sì	Sì	No	No	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Migration Hub Orchestratore	Sì	Sì	No	Sì	Sì	Sì
AWS Migration Hub Refactor Spaces	Sì	Sì	Sì	Sì	Sì	Sì
AWS Migration Hub Strategy Recommendations	Sì	No	No	No	Sì	Sì
Amazon Monitron	Sì	Sì	No	Sì	Sì	Sì
Amazon MQ	Sì	Sì	No	Sì	Sì	Sì
Amazon Neptune	Sì	Sì	No	No	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Analisi di Amazon Neptune	Sì	Sì	No	Sì	Sì	No
AWS Network Firewall	Sì	Sì	No	Sì	Sì	Sì
AWS Network Manager	Sì	Sì	No	Sì	Sì	Sì(Informazioni)
AWS Network Manager Chat	Sì	No	No	No	Sì	No
Amazon Nimble Studio	Sì	Sì	No	Sì	Sì	No
Amazon One Enterprise	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
OpenSearchIngestione di Amazon	Sì	Sì	No	Sì	Sì	Sì
Amazon OpenSearch Serverless	Sì	Sì	No	Sì	Sì	Sì
OpenSearch Servizio Amazon	Sì	Sì	Sì	Sì	Sì	Sì
AWS OpsWorks	Sì	Sì	No	No	Sì	No
AWS OpsWorks Gestione della configurazione	Sì	Sì	No	No	Sì	No
AWS Organizations	Sì	Sì	No	Sì	No	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Outposts	Sì	Sì	No	Sì	Sì	Sì
AWS Panorama	Sì	Sì	No	Sì	Sì	Sì
AWS Partner Gestione centralizzata degli account	Sì	No	No	No	Sì	No
AWS Payment Cryptography	Sì	Sì	No	Sì	Sì	No
AWS Pagamenti	Sì	No	No	No	Sì	No
AWS Approfondimenti sulle prestazioni	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Personalize	Sì	Sì	No	No	Sì	No
Amazon Pinpoint	Sì	Sì	No	Sì	Sì	No
Servizio e-mail Amazon Pinpoint	Sì	Sì	No	Sì	Sì	No
Servizio di SMS e messaggi vocali Amazon Pinpoint	Sì	No	No	No	Sì	No
Amazon Pinpoint SMS and Voice Service v2	Sì	Sì	No	Sì	Sì	No
Amazon Polly	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Listino prezzi AWS	Sì	No	No	No	Sì	No
AWS 5G privato	Sì	Sì	No	Sì	Sì	No
AWS Private CA Connettore per Active Directory	Sì	Sì	No	Sì	Sì	No
AWS Private CA Connettore per SCEP	Sì	Sì	No	Sì	Sì	No
AWS Private Certificate Authority (AWS Private CA)	Sì	Sì	Sì	Sì	Sì	No
AWS Proton	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Console per gli ordini di acquisto	Sì	Sì	No	Sì	Sì	No
Amazon Q Business	Sì	Sì	No	Sì	Sì	Sì
App Amazon Q Business Q	Sì	Sì	No	No	Sì	No
Sviluppatore Amazon Q	Sì	No	No	No	Sì	Sì
Amazon Q in Connect	Sì	Sì	No	Sì	Sì	No
Database Amazon Quantum Ledger (Amazon QLDB)	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon QuickSight	 Sì	 Sì	 No	 Sì	 Sì	 No
API dati di Amazon RDS	 Sì	 Sì	 No	 No	 Sì	 No
Amazon RDS IAM Authentication	 Sì	 Sì	 No	 No	 Sì	 No
AWS Cestino di riciclaggio	 Sì	 Sì	 No	 Sì	 Sì	 No
Amazon Redshift	 Sì	 Sì	 No	 Sì	 Sì	 Sì
API dati di Amazon Redshift	 Sì	 Sì	 No	 No	 Sì	 No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Redshift Serverless	Sì	Sì	Sì	Sì	Sì	No
Amazon Rekognition	Sì	Sì	Parziale (Informazioni)	Sì	Sì	No
Amazon Relational Database Service (Amazon RDS) (Info)	Sì	Sì	No	Sì	Sì	Sì
AWS re:Post Privata	Sì	Sì	No	Sì	Sì	Sì
AWS Resilience Hub	Sì	Sì	No	Sì	Sì	No
AWS Resource Access Manager (AWS RAM)	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Esploratore di risorse AWS	Sì	Sì	No	Sì	Sì	Sì
AWS Resource Groups	Sì	Sì	No	Sì	Parzia (Informazioni)	No
AWS Resource Groups Tagging API	Sì	No	No	No	Sì	No
Amazon RHEL Knowledgebase Portal	Sì	No	No	No	Sì	No
AWS RoboMaker	Sì	Sì	No	Sì	Sì	Sì
Amazon Route 53	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Route 53 Application Recovery Controller - Spostamento zonale	Sì	Sì	No	No	Sì	No
Domini Amazon Route 53	Sì	No	No	No	No	No
Profili Amazon Route 53	Sì	Sì	No	Sì	Sì	No
Cluster di ripristino di Amazon Route 53	Sì	Sì	No	No	Sì	No
Configurazione dei controlli di ripristino Amazon Route 53	Sì	Sì	No	Sì	Sì	No
Preparazione al ripristino di Amazon Route 53	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Route 53 Resolver	Sì	Sì	No	Sì	Sì	Sì
Amazon S3 Express	Sì	Sì	No	No	Sì	No
Amazon S3 Glacier	Sì	Sì	Sì	Sì	Sì	No
Amazon SageMaker	Sì	Sì	No	Sì	Sì	Parziale(Informazioni)
Funzionalità SageMaker geospaziali di Amazon	Sì	Sì	No	Sì	Sì	No
Amazon SageMaker Ground Truth sintetico	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Savings Plans	Sì	Sì	No	Sì	Sì	No
AWS Secrets Manager	Sì	Sì	Sì	Sì	Sì	No
AWS Security Hub	Sì	Sì	No	Sì	Sì	Sì
Amazon Security Lake	Sì	Sì	No	No	Sì	Sì
AWS Security Token Service (AWS STS)	Sì	Parziale (Informazioni)	No	Sì	Parziale (Informazioni)	No
AWS Serverless Application Repository	Sì	Sì	Sì	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Service Catalog	Sì	Sì	No	Sì	Sì	Sì
Service Quotas	Sì	Sì	No	Sì	Sì	No
AWS Shield	Sì	Sì	No	Sì	Sì	Sì
AWS Signer	Sì	Sì	Sì	Sì	Sì	No
AWS Accedi	Sì	No	No	No	Sì	No
Amazon SimpleDB	Sì	Sì	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Amazon Simple Email Service - Gestione della posta	 Sì	 Sì	 No	 Sì	 Sì	 Sì
Amazon Simple Email Service (Amazon SES) v2	 Sì	 Parzia (Informazioni)	 Sì	 Sì	 Parzia (Informazioni)	 Sì
Amazon Simple Notification Service (Amazon SNS)	 Sì	 Sì	 Sì	 Sì	 Sì	 No
Amazon Simple Queue Service (Amazon SQS)	 Sì	 Sì	 Sì	 Parziale	 Sì	 No
Amazon Simple Storage Service (Amazon S3)	 Sì	 Sì	 Sì	 Parziale (Informazioni)	 Sì	 Parziale (Informazioni)

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Oggetto Amazon Simple Storage Service (Amazon S3) Lambda	Sì	Sì	No	No	Sì	No
Amazon Simple Storage Service (Amazon S3) su AWS Outposts	Sì	Sì	Sì	No	Sì	Sì
Amazon Simple Workflow Service (Amazon SWF)	Sì	Sì	No	Sì	Sì	No
AWS SimSpaceWeaver	Sì	Sì	No	Sì	Sì	No
AWS Site-to-Site VPN	Sì	Sì	No	No	Sì	Sì
AWS Snowball	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Snowball Edge	Sì	No	No	No	Sì	No
AWS Snow Device Management	Sì	Sì	No	Sì	Sì	No
AWS SQL Workbench	Sì	Sì	No	Sì	Sì	No
AWS Step Functions	Sì	Sì	No	Sì	Sì	No
AWS Storage Gateway	Sì	Sì	No	Sì	Sì	No
Catena di approvvigionamento di AWS	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Support App in Slack	 Sì	 No	 No	 No	 Sì	 No
AWS Support	 Sì	 No	 No	 No	 Sì	 Sì
AWS Support Piani	 Sì	 No	 No	 No	 Sì	 No
AWS Support Raccomandazioni	 Sì	 No	 No	 No	 Sì	 No
AWS Sostenibilità	 Sì	 No	 No	 No	 Sì	 No
AWS Systems Manager	 Sì	 Sì	 No	 Sì	 Sì	 Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Systems Manager per SAP	Sì	Sì	No	Sì	Sì	No
AWS Systems Manager GUI Connect	Sì	No	No	No	Sì	No
AWS Systems Manager Incident Manager	Sì	Sì	Sì	Sì	Sì	Sì
AWS Systems Manager Incident Manager Contatti	Sì	Sì	Sì	No	Sì	No
Editor di tag	Sì	No	No	No	Sì	No
AWS Impostazioni fiscali	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Telco Network Builder	Sì	Sì	No	Sì	Sì	No
Amazon Textract	Sì	No	No	No	Sì	No
Amazon Timestream	Sì	Sì	No	Sì	Sì	No
Amazon Timestream Influxdb	Sì	Sì	No	Sì	Sì	Sì
AWS API Tiros (per Reachability Analyzer)	Sì	No	No	No	No	No
Amazon Transcribe	Sì	Sì	No	Sì	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS Transfer Family	Sì	Sì	No	Sì	Sì	No
Amazon Translate	Sì	Sì	No	Sì	Sì	No
AWS Trusted Advisor	Parziale (Informazioni)	Sì	No	No	Parziale	Sì
AWS Notifiche per gli utenti	Sì	Sì	No	Sì	Sì	Sì
AWS Notifiche utente e contatti	Sì	Sì	No	Sì	Sì	No
AWS Sottoscrizioni degli utenti	Sì	No	No	No	Sì	No

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Accesso verificato da AWS	Sì	No	No	No	Sì	No
Autorizzazioni verificate da Amazon	Sì	Sì	No	No	Sì	No
Amazon Virtual Private Cloud (Amazon VPC)	Sì	Parziale (Informazioni)	Parziale (Informazioni)	Sì	Sì	Parziale (Informazioni)
Amazon VPC Lattice	Sì	Sì	No	Sì	Sì	No
Servizi Amazon VPC Lattice	Sì	Sì	No	No	Sì	No
AWS WAF	Sì	Sì	No	Sì	Sì	Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
AWS WAF Classic	 Sì	 Sì	 No	 Sì	 Sì	 Sì
AWS WAF Regionale	 Sì	 Sì	 No	 Sì	 Sì	 Sì
AWS Well-Architected Tool	 Sì	 Sì	 No	 Sì	 Sì	 No
AWS Wickr	 Sì	 Sì	 No	 Sì	 Sì	 No
Amazon WorkDocs	 Sì	 No	 No	 No	 Sì	 No
Amazon WorkMail	 Sì	 Sì	 No	 Sì	 Sì	 Sì

Servizio	Azioni	Autorizzazioni a livello di risorsa	Policy basate su risorse	ABAC	Credenziali temporanee	Ruoli collegati al servizio
Flusso di WorkMail messaggi Amazon	Sì	Sì	No	No	Sì	No
Amazon WorkSpaces	Sì	Sì	No	Sì	Sì	No
Browser WorkSpaces sicuro Amazon	Sì	Sì	No	Sì	Sì	Sì
Amazon WorkSpaces Thin Client	Sì	Sì	No	Sì	Sì	No
AWS X-Ray	Sì	Parziale (Informazioni)	No	Parziale (Informazioni)	Sì	No

Ulteriori informazioni

Amazon CloudFront

CloudFront non ha ruoli collegati ai servizi, ma Lambda @Edge sì. Per ulteriori informazioni, consulta Service-Linked Roles for Lambda @Edge nella Amazon Developer Guide. CloudFront

AWS CloudTrail

CloudTrail supporta politiche basate sulle risorse solo sui CloudTrail canali utilizzati per le integrazioni di CloudTrail Lake con fonti di eventi esterne a. AWS

CloudTrail supporta il controllo degli accessi basato su tag per gli archivi di dati e i canali di eventi CloudTrail Lake. CloudTrail non supporta i controlli di accesso basati su tag per i sentieri.

Amazon CloudWatch

CloudWatch i ruoli collegati ai servizi non possono essere creati utilizzando la AWS Management Console funzionalità Alarm Actions e supportano solo la funzionalità Alarm Actions.

AWS CodeBuild

CodeBuild supporta la condivisione di risorse tra account utilizzando. AWS RAM

CodeBuild supporta ABAC per le azioni basate su progetti.

AWS Config

AWS Config supporta le autorizzazioni a livello di risorsa per l'aggregazione e le regole di dati multiaccount e più regioni. AWS Config Per un elenco di risorse supportate, consulta la sezione Aggregazione di dati multi-regione multi-account e la sezione Regole AWS Config della AWS Config Guida alle API.

AWS Database Migration Service

È possibile creare e modificare le policy allegate alle chiavi di AWS KMS crittografia create per crittografare i dati migrati verso gli endpoint di destinazione supportati. Gli endpoint di destinazione supportati includono Amazon Redshift e Amazon S3. Per ulteriori informazioni, consulta Creazione e utilizzo di AWS KMS chiavi per crittografare i dati di destinazione di Amazon Redshift e AWS KMS Creazione di chiavi per crittografare oggetti di destinazione Amazon S3 nella Guida per l'utente.AWS Database Migration Service

Amazon Elastic Compute Cloud

I ruoli collegati ai servizi EC2 possono essere utilizzati solo per le seguenti funzionalità: Richieste di istanze Spot, Richieste di serie di istanze spot, Parchi istanze Amazon EC2 e Avvio rapido di istanze Windows.

Amazon Elastic Container Service

Solo alcune operazioni Amazon ECS supportano le autorizzazioni a livello di risorse.

AWS Elemental MediaPackage

MediaPackage supporta ruoli collegati ai servizi per la pubblicazione dei log di accesso dei clienti ma non per CloudWatch altre azioni API.

AWS Identity and Access Management

IAM supporta solo un tipo di policy basata su risorse detta policy di attendibilità del ruolo, collegata a un ruolo IAM. Per ulteriori informazioni, consulta Concessione di autorizzazioni agli utenti per il cambio di ruoli.

IAM supporta il controllo degli accessi basato su tag per la maggior parte delle risorse IAM. Per ulteriori informazioni, consulta Tagging delle risorse IAM.

Solo alcune delle operazioni API per IAM possono essere chiamate con credenziali temporanee. Per ulteriori informazioni, consulta la sezione di confronto delle opzioni API.

AWS IoT

I dispositivi collegati AWS IoT vengono autenticati utilizzando certificati X.509 o utilizzando Amazon Cognito Identities. Puoi allegare AWS IoT policy a un certificato X.509 o Amazon Cognito Identity per controllare ciò a cui il dispositivo è autorizzato a fare. Per ulteriori informazioni, consulta Sicurezza e identità per AWS IoT nella Guida per gli sviluppatori di AWS IoT .

AWS Lambda

Lambda supporta il controllo degli accessi basato su attributi (ABAC) per le azioni API che utilizzano una funzione Lambda come risorsa richiesta. I livelli, gli strumenti di mappatura dell'origine degli eventi e le risorse di configurazione della firma del codice non sono supportati.

Lambda non dispone di ruoli collegati al servizio, a differenza di Lambda@Edge. Per ulteriori informazioni, consulta Service-Linked Roles for Lambda @Edge nella Amazon Developer Guide. CloudFront

Amazon Lightsail

Lightsail supporta parzialmente le autorizzazioni a livello di risorsa e ABAC. Per ulteriori informazioni, consulta la sezione Operazioni, risorse e chiavi della condizione di Amazon Lightsail.

Amazon Managed Streaming per Apache Kafka (MSK)

Puoi allegare una policy del cluster a un cluster Amazon MSK configurato per la connettività multi-VPC.

AWS Network Manager

AWS Cloud WAN supporta anche ruoli collegati ai servizi. Per ulteriori informazioni, consulta i ruoli collegati ai servizi AWS Cloud WAN nella Amazon VPC AWS Cloud WAN Guide.

Amazon Relational Database Service

Amazon Aurora è un motore di database relazionale completamente gestito compatibile con MySQL e PostgreSQL. Quando si configurano nuovi server di database mediante Amazon RDS, come motore di database è possibile scegliere Aurora MySQL o Aurora PostgreSQL. Per ulteriori informazioni, consulta la sezione Gestione di identità e accessi per Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Amazon Rekognition

Le policy basate sulle risorse sono supportate solo per la copia dei modelli delle etichette personalizzate Amazon Rekognition.

AWS Resource Groups

Gli utenti possono assumere un ruolo con una policy che consente operazioni con Resource Groups.

Amazon SageMaker

I ruoli collegati ai servizi sono attualmente disponibili per i lavori in SageMaker Studio e di SageMaker formazione.

AWS Security Token Service

AWS STS non dispone di «risorse», ma consente di limitare l'accesso in modo analogo agli utenti. Per ulteriori informazioni, consulta Rifiutare l'accesso alle credenziali di sicurezza temporanee tramite il nome.

Solo alcune delle operazioni API per AWS STS supportare le chiamate con credenziali temporanee. Per ulteriori informazioni, consulta la sezione di confronto delle opzioni API.

Amazon Simple Email Service

Puoi utilizzare solo le autorizzazioni a livello di risorsa in dichiarazioni di policy che fanno riferimento a operazioni correlate all'invio di e-mail, ad esempio `ses:SendEmail` o `ses:SendRawEmail`. Per le dichiarazioni di policy che fanno riferimento a qualsiasi altra operazione, l'elemento Resource può contenere solo `*`.

Solo l'API Amazon SES supporta le credenziali di sicurezza temporanee. L'interfaccia SMTP Amazon SES non supporta credenziali SMTP derivate da credenziali di sicurezza temporanee.

Amazon Simple Storage Service

Amazon S3 supporta l'autorizzazione basata su tag solo per le risorse di oggetti.

Amazon S3 supporta i ruoli collegati ai servizi per Amazon S3 Storage Lens.

AWS Trusted Advisor

L'accesso all'API Trusted Advisor avviene tramite l' AWS Support API ed è controllato dalle politiche AWS Support IAM.

Amazon Virtual Private Cloud

In una policy utente IAM, non puoi limitare le autorizzazioni a un endpoint VPC Amazon specifico. Qualsiasi elemento `Action` che include le operazioni API `ec2:*VpcEndpoint*` o `ec2:DescribePrefixLists` deve specificare `"'Resource": "*"'`. Per ulteriori informazioni, consulta la sezione Gestione delle identità e degli accessi per endpoint VPC e servizi endpoint VPC nella Guida di AWS PrivateLink .

Amazon VPC supporta il collegamento di una singola policy di risorse a un endpoint VPC per limitare l'accesso tramite tale endpoint. Per ulteriori informazioni sull'utilizzo di policy basate su risorse per controllare l'accesso alle risorse da specifici endpoint Amazon VPC, consulta la sezione Controllo dell'accesso ai servizi tramite policy di endpoint nella Guida di AWS PrivateLink .

Amazon VPC non ha ruoli collegati ai servizi, ma sì. AWS Transit Gateway Per ulteriori informazioni, consulta Use service-linked roles for transit gateway nella Amazon VPC Guide. AWS Transit Gateway

AWS X-Ray

X-Ray non supporta le autorizzazioni a livello di servizio per tutte le operazioni.

X-Ray supporta il controllo degli accessi basato su tag per gruppi e regole di campionamento.

Firma AWS delle richieste API

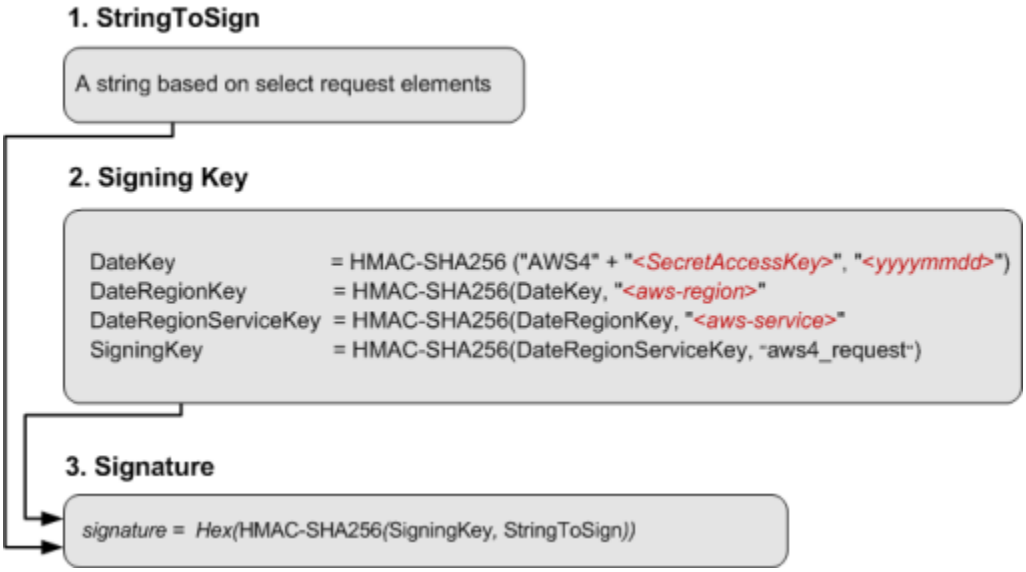
> **Important**
>
> Se utilizzi uno strumento AWS SDK (vedi Codice di esempio e librerie) o a riga di AWS comando (CLI) a cui inviare richieste AWS API, puoi saltare questa sezione perché i client SDK e CLI autenticano le tue richieste utilizzando le chiavi di accesso che fornisci. A meno che tu non abbia una buona ragione per non farlo, ti consigliamo di utilizzare sempre un SDK o una CLI.
> Nelle regioni che supportano più versioni di firma, per le richieste di firma manuale è necessario specificare quale versione della firma viene utilizzata. Quando utilizzi punti di accesso multi-regione, gli SDK e le CLI vengono commutate automaticamente per utilizzare Signature Version 4A senza ulteriori configurazioni.

Le informazioni di autenticazione inviate in una richiesta devono includere una firma. Per calcolare una firma, devi prima concatenare gli elementi della richiesta per formare una stringa, denominata stringa da firmare. Si utilizza quindi una chiave di firma per calcolare il codice di autenticazione dei messaggi basato su hash (HMAC) della stringa da firmare.

Nella versione 4 di AWS Signature, non usi la tua chiave di accesso segreta per firmare la richiesta. Invece, si usa prima la propria chiave di accesso segreta per generare una chiave di firma. La chiave di firma derivata è specifica per data, servizio e regione. Per ulteriori informazioni su come ottenere una chiave di firma in diversi linguaggi di programmazione, vedi Richiesta di esempi di firma.

Signature Version 4 è il protocollo di AWS firma. AWS supporta anche un'estensione, Signature Version 4A, che supporta le firme per le richieste API multiregionali. Per ulteriori informazioni, consulta il progetto a-signing-examples sigv4 su. GitHub

Il diagramma seguente illustra il processo generale di calcolo di una firma.



- La stringa da firmare dipende dal tipo di richiesta. Ad esempio, quando utilizzi l'intestazione dell'autorizzazione HTTP o i parametri della query per l'autenticazione, utilizzi una combinazione variabile di elementi della richiesta per creare la stringa da firmare. Per una richiesta HTTP POST, la policy `POST` nella richiesta è la stringa che firmi.
- Per chiave di firma, il diagramma mostra una serie di calcoli, dove il risultato di ogni passaggio viene inserito nel passaggio successivo. Il passaggio finale è la chiave di firma.
- Quando un AWS servizio riceve una richiesta autenticata, ricrea la firma utilizzando le informazioni di autenticazione contenute nella richiesta. Se le firme corrispondono, il servizio elabora la richiesta. In caso contrario, la richiesta viene respinta.

Indice

Quando firmare le richieste

Quando scrivi codice personalizzato che invia richieste API a AWS, devi includere il codice che firma le richieste. Potrebbe essere necessario scrivere codice personalizzato perché:

- Utilizzi un linguaggio di programmazione per il quale non esiste un SDK AWS .
- È necessario il controllo completo su come vengono inviate le richieste AWS.

Perché le richieste vengono firmate

Il processo di firma aiuta a proteggere le richieste, poiché consente di:

- Verificare l'identità del richiedente

  Le richieste autenticate richiedono una firma che hai creato utilizzando le tue chiavi di accesso (ID chiave di accesso, chiave di accesso segreta). Se stai utilizzando credenziali di sicurezza temporanee, i calcoli della firma richiedono anche un token di sicurezza. Per ulteriori informazioni, consulta AWS accesso programmatico con credenziali di sicurezza.

- Proteggere i dati in transito

  Per evitare che una richiesta venga modificata mentre è in transito, alcuni elementi della richiesta stessa vengono utilizzati per calcolare un hash (digest) e il valore hash risultante è incluso come parte della richiesta. Quando un utente Servizio AWS riceve la richiesta, utilizza le stesse informazioni per calcolare un hash e lo confronta con il valore hash della richiesta. Se i valori non corrispondono, AWS nega la richiesta.

- Garantire la protezione da possibili attacchi di tipo replay

  Nella maggior parte dei casi, una richiesta deve pervenire AWS entro cinque minuti dalla data indicata nella richiesta. In caso contrario, AWS nega la richiesta.

Elementi della firma di una richiesta AWS API

> **Important**
>
> A meno che non utilizzi gli AWS SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo delle AWS firme nella

versione 4 di Signature può essere un'impresa complessa e ti consigliamo di utilizzare gli AWS SDK o la CLI ogni volta che è possibile.

Ogni richiesta HTTP/HTTPS che utilizza Signature Version 4 deve contenere questi elementi.

Elementi

- Specifica dell'endpoint
- Azione
- Parametri dell'operazione
- Data
- Informazioni sull'autenticazione

Specifica dell'endpoint

Specifica il nome DNS dell'endpoint a cui inviare la richiesta. Questo nome di solito contiene il codice del servizio e la regione. Ad esempio, l'endpoint Amazon DynamoDB per la regione `us-east-1` è `dynamodb.us-east-1.amazonaws.com`.

Per le richieste HTTP/1.1 devi includere l'intestazione `Host`. Per le richieste HTTP/2, puoi utilizzare l'intestazione `:authority` o l'intestazione `Host`. Utilizza solo l'intestazione `:authority` per conformità con la specifica HTTP/2. Non tutti i servizi supportano le richieste HTTP/2.

Per gli endpoint supportati da ciascun servizio, consulta Endpoint e quote del servizio nella Riferimenti generali di AWS.

Azione

Specifica un'operazione API per il servizio. Ad esempio, l'operazione `CreateTable` di DynamoDB o l'operazione `DescribeInstances` di Amazon EC2.

Per le operazioni supportate da ciascun servizio, consulta la Guida di riferimento per l'autorizzazione del servizio.

Parametri dell'operazione

Specifica i parametri per l'operazione specificata nella richiesta. Ogni azione AWS API ha una serie di parametri obbligatori e opzionali. La versione dell'API di solito è un parametro obbligatorio.

Per i parametri supportati da un'operazione API, consulta la Documentazione di riferimento delle API per il servizio.

Data

Specifica la data e l'ora della richiesta. Con la data e l'ora nella richiesta puoi evitare che le terze parti intercettino la tua richiesta e la inviino in un secondo momento. La data specificata nell'ambito delle credenziali deve corrispondere alla data della richiesta.

Il timestamp deve essere in UTC e deve avere il seguente formato ISO 8601: YYYYMMDDTHHMMSSZ. Ad esempio, `20220830T123600Z`. Non includere i millisecondi nel time stamp.

Si può utilizzare un'intestazione `date`, un'intestazione `x-amz-date` o includere `x-amz-date` come parametro di query. Se non riusciamo a trovare un'intestazione `x-amz-date`, cerchiamo un'intestazione `date`.

Informazioni sull'autenticazione

Ogni richiesta inviata deve includere le seguenti informazioni. AWS utilizza queste informazioni per garantire la validità e l'autenticità della richiesta.

- Algoritmo: utilizza `AWS 4-HMAC-SHA256` per specificare Signature Version 4 con l'algoritmo hash `HMAC-SHA256`.
- Credenziale: una stringa composta dall'ID della chiave di accesso, dalla data in formato AAAAMMGG, dal codice della regione, dal codice del servizio e dalla stringa di chiusura `aws4_request`, separati da barre (/). Devi utilizzare caratteri minuscoli per la regione, il codice del servizio e la stringa di chiusura.

  ```
  AKIAIOSFODNN7EXAMPLE/YYYYMMDD/region/service/aws4_request
  ```

- Intestazioni firmate: le intestazioni HTTP da includere nella firma, separate da punto e virgola (;). Ad esempio, `host;x-amz-date`.
- Firma: una stringa con codifica esadecimale che rappresenta la firma calcolata. Devi calcolare la firma utilizzando l'algoritmo specificato nel parametro `Algorithm`.

Metodi di autenticazione

> **Important**
>
> A meno che non utilizzi gli AWS SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo delle AWS firme nella versione 4 di Signature può essere un'impresa complessa e ti consigliamo di utilizzare gli AWS SDK o la CLI ogni volta che è possibile.

Puoi esprimere le informazioni di autenticazione utilizzando uno dei seguenti metodi.

Intestazione HTTP di autorizzazione

L'intestazione `Authorization` HTTP è il metodo più comune per autenticare una richiesta. Tutte le operazioni del REST API (ad eccezione dei caricamenti basati su browser che utilizzano richieste `POST`) richiedono questa intestazione. Per ulteriori informazioni sul valore dell'intestazione di autorizzazione e su come calcolare la firma e le opzioni correlate, consulta Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) nel riferimento all'API di Amazon S3.

Di seguito è riportato un esempio del valore per l'intestazione `Authorization`. Le interruzioni di riga vengono aggiunte a questo esempio solo per migliorare la leggibilità. Nel tuo codice, l'intestazione deve essere una stringa continua. Non c'è una virgola tra l'algoritmo e le credenziali, ma gli altri elementi devono essere separati da virgole.

```
Authorization: AWS 4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

La tabella seguente descrive i vari componenti del valore dell'intestazione di autorizzazione nell'esempio precedente:

Componente	Descrizione
Autorizzazione	L'algoritmo utilizzato per calcolare la firma. È necessario fornire questo valore quando si utilizza AWS Signature Version 4 per l'autenti

Componente	Descrizione
	cazione. La stringa specifica Signature Version 4 () e l'algoritmo di AWS firma (). `AWS 4 HMAC-SHA256`
Credential	L'ID della chiave di accesso e le informazioni sull'ambito, che includono la data, la regione e il servizio utilizzati per calcolare la firma. Questa stringa ha il seguente formato: `<your-access-key-id>/<date>/ <aws-region>/<aws-service>/ aws4_request` Dove: il valore `<date>` è specificato utilizzando il formato AAAAMMGG. Il valore `<aws-service>` è s3 quando si invia una richiesta ad Amazon S3.
SignedHeaders	Un elenco separato da punto e virgola delle intestazioni di richiesta che hai utilizzato per il calcolo `Signature` . L'elenco include solo i nomi delle intestazioni e i nomi delle intestazioni devono essere in minuscolo. Ad esempio: host; range; x-amz-date
Firma	La firma a 256 bit espressa come 64 caratteri esadecimali minuscoli. Ad esempio:`fe5f80f77d5fa3beca 038a248ff027d0445342fe2855d dc963176630326f1024` Tieni presente che i calcoli della firma variano a seconda dell'opzione scelta per trasferire il payload.

Parametri della stringa di query

È possibile utilizzare una stringa di query per esprimere una richiesta interamente in un URL. In questo caso, utilizzi i parametri di interrogazione per fornire le informazioni sulla richiesta, incluse le informazioni di autenticazione. Poiché la firma richiesta è parte dell'URL, questo tipo di URL viene chiamato URL pre-firmato. Puoi utilizzare gli URL preimpostati per incorporare link cliccabili in HTML; questi link saranno validi per un massimo di sette giorni. Per ulteriori informazioni, consulta Autenticazione delle richieste: utilizzo dei parametri di interrogazione (AWS Signature versione 4) nel riferimento all'API Amazon S3.

Di seguito è riportato un esempio di URL prefirmato. Le interruzioni di riga vengono aggiunte a questo esempio solo per migliorare la leggibilità.

```
https://s3.amazonaws.com/examplebucket/test.txt ?
X-Amz-Algorithm=AWS4-HMAC-SHA256 &
X-Amz-Credential=<your-access-key-id>/20130721/us-east-1/s3/aws4_request &
X-Amz-Date=20130721T201207Z &
X-Amz-Expires=86400 &
X-Amz-SignedHeaders=host &X-Amz-Signature=<signature-value>
```

> **Note**
>
> Il valore `X-Amz-Credential` nell'URL mostra il carattere "/" solo per leggibilità. In pratica, dovrebbe essere codificato come %2F. Per esempio: `&X-Amz-Credential=<your-access-key-id>%2F20130721%2Fus-east-1%2Fs3%2Faws4_request`

La tabella seguente descrive i parametri della query nell'URL che forniscono informazioni di autenticazione.

Nome parametro stringa di query	Descrizione
Algoritmo X-Amz	Identifica la versione di AWS Signature e l'algoritmo utilizzato per calcolare la firma. Per la versione 4 di AWS Signature, imposti questo valore del parametro su. `AWS 4-HMAC-SHA256` Questa stringa identifica AWS

Nome parametro stringa di query	Descrizione
	Signature Version 4 (AWS 4) e l'algoritmo HMAC-SHA256 (`HMAC-SHA256`).
Credenziali X-Amz	Oltre all'ID della chiave di accesso, questo parametro fornisce anche l'ambito (AWS regione e servizio) per il quale la firma è valida. Questo valore deve corrispondere all'ambito utilizzato nei calcoli della firma, illustrato nella sezione seguente. La forma generale per questo valore di parametro è la seguente: `<your-access-key-id>/<date>/<AWS Region>/<AWS-service>/aws4_request` Ad esempio: `AKIAIOSFODNN7EXAMPLE/20130721/us-east-1/s3/aws4_request` Per un elenco di stringhe AWS regionali, consulta Endpoint regionali nel Riferimento AWS generale.
X-Amz-Date	Il formato di data e ora deve seguire lo standard ISO 8601 e deve essere formattato con il formato `yyyyMMddTHHmmssZ`. Ad esempio, se la data e l'ora erano "08/01/2016 15:32:41.982-700", devono prima essere convertite in UTC (Coordinated Universal Time) e quindi inviate come "20160801T223241Z".

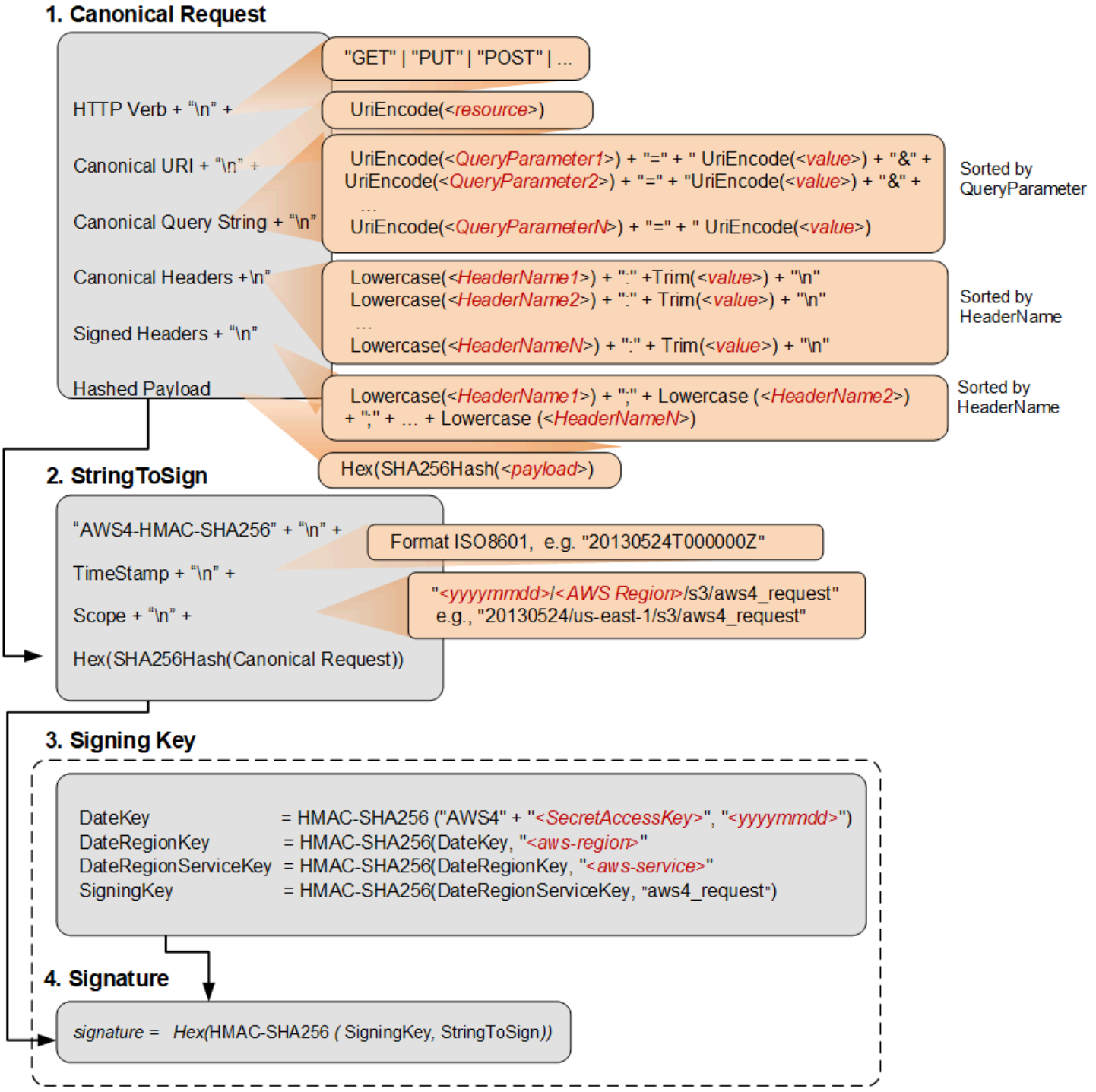
Nome parametro stringa di query	Descrizione
X-Amz-Expires	Fornisce il periodo di tempo, in secondi, per il quale l'URL prefirmato generato è valido. Ad esempio, 86400 (24 ore). Questo valore è un numero intero. Il valore minimo che puoi impostare è 1 e il massimo è 604800 (sette giorni). Un URL prefirmato può essere valido per massimo sette giorni perché la chiave di firma utilizzata nel calcolo della firma è valida per un massimo di sette giorni.
X-Amz- SignedHeaders	Elenca le intestazioni che hai utilizzato per calcolare la firma. Per i calcoli delle firme sono necessarie le seguenti intestazioni: • L'intestazione dell'host HTTP. • Qualsiasi intestazione x-amz-* che intendi aggiungere alla richiesta. Per maggiore sicurezza, devi firmare tutte le intestazioni della richiesta che intendi includere nella richiesta.
X-Amz-Signature	Fornisce la firma per autenticare la richiesta . Questa firma deve corrispondere alla firma calcolata dal servizio; in caso contrario, il servizio rifiuta la richiesta. Ad esempio, `733255ef022bec3f2a8701cd61d 4b371f3f28c9f193a1f02279211 d48d5193d7` I calcoli delle firme sono descritti nella sezione seguente.

Nome parametro stringa di query	Descrizione
X-Amz-Security-Token	Parametro opzionale delle credenziali se si utilizzano credenziali provenienti dal servizio STS.

Creare una richiesta AWS API firmata

> **Important**
>
> Se utilizzi uno strumento AWS SDK (vedi Codice di esempio e librerie) o a riga di AWS comando (CLI) a cui inviare richieste AWS API, puoi saltare questa sezione perché i client SDK e CLI autenticano le tue richieste utilizzando le chiavi di accesso che fornisci. A meno che tu non abbia una buona ragione per non farlo, ti consigliamo di utilizzare sempre un SDK o una CLI.
> Nelle regioni che supportano più versioni di firma, per le richieste di firma manuale è necessario specificare quale versione della firma viene utilizzata. Quando utilizzi punti di accesso multi-regione, gli SDK e le CLI vengono commutate automaticamente per utilizzare Signature Version 4A senza ulteriori configurazioni.

Di seguito è riportata una panoramica del processo per creare una richiesta firmata. Per calcolare una firma, occorre una stringa da firmare. Quindi calcoli un hash HMAC-SHA256 della stringa da firmare utilizzando una chiave di firma. Il diagramma seguente illustra il processo, inclusi i vari componenti della stringa che hai creato per la firma.



La tabella seguente descrive le funzioni mostrate nel diagramma. Per queste funzioni devi implementare il codice. [Per ulteriori informazioni, consulta gli esempi di codice negli SDK. AWS]

Funzione	Descrizione
`Lowercase()`	Converte la stringa in minuscolo.
`Hex()`	Codifica in base 16 minuscola.
`SHA256Hash()`	Funzione hash crittografica Secure Hash Algorithm (SHA).
`HMAC-SHA256()`	Calcola HMAC utilizzando l'algoritmo SHA256 con la chiave di firma fornita. Questa è la firma definitiva.
`Trim()`	Rimuove eventuali spazi bianchi all'inizio o alla fine della stringa.
`UriEncode()`	L'URI codifica ogni byte. UriEncode() deve applicare le seguenti regole: • L'URI codifica ogni byte tranne i caratteri senza riserve: 'A'-'Z', 'a'-'z', '0'-'9', '-', ' . ','_' e '~'. • Il carattere di spazio è un carattere riservato e deve essere codificato come "%20" (e non come "+"). • Ogni byte codificato in URI è formato da una '%' e dal valore esadecimale a due cifre del byte. • Le lettere nel valore esadecimale devono essere maiuscole, ad esempio "%1A". • Codifica la barra, '/', ovunque tranne nel nome della chiave dell'oggetto. Ad esempio, se il nome della chiave dell'oggetto è `photos/Jan/sample.jpg` , la barra nel nome della chiave non è codificata.

Funzione	Descrizione
	Important Le UriEncode funzioni standard fornite dalla piattaforma di sviluppo potrebbero non funzionare a causa delle differenz e di implementazione e della relativa ambiguità nelle RFC sottostanti. Ti consigliamo di scrivere una UriEncode funzione personalizzata per assicurarti che la codifica funzioni. Per vedere un esempio di UriEncode funzione in Java, consulta Java Utilities sul GitHub sito Web.

Note

Quando firmi le tue richieste, puoi utilizzare AWS Signature Version 4 o AWS Signature Version 4A. La differenza fondamentale tra le due versioni è determinata dalla modalità di calcolo della firma. Con AWS Signature Version 4A, la firma non include informazioni specifiche della regione e viene calcolata utilizzando l'algoritmo. `AWS 4-ECDSA-P256-SHA256`

Credenziali di sicurezza temporanee

Invece di utilizzare credenziali a lungo termine per firmare una richiesta, è possibile utilizzare credenziali di sicurezza temporanee fornite da (). AWS Security Token Service AWS STS

Quando si utilizzano credenziali di sicurezza temporanee, è necessario aggiunge `X-Amz-Security-Token` all'intestazione di autorizzazione o alla stringa di query per contenere il token di sessione. Alcuni servizi richiedono l'aggiunta di `X-Amz-Security-Token` alla richiesta canonica. Per gli altri servizi, aggiungi il parametro `X-Amz-Security-Token` alla fine, dopo aver calcolato la firma. Per i dettagli, consulta la documentazione relativa a ciascuna Servizio AWS di esse.

Riepilogo delle fasi di firma

Fase 1: creazione di una richiesta canonica

Disponi i contenuti della tua richiesta (host, operazione, intestazioni, ecc.) in un formato standard (canonico). La richiesta canonica è uno degli input utilizzati per creare una stringa di firma. Per informazioni dettagliate, vedi Elementi della firma di una richiesta AWS API.

Fase 2: creazione di un hash della richiesta canonica

Ricava una chiave di firma eseguendo una serie di operazioni hash con chiave (operazioni HMAC) nella data, nella regione e nel servizio della richiesta, utilizzando la chiave di accesso AWS segreta come chiave per l'operazione di hashing iniziale.

Fase 3: creazione di una stringa da firmare

Crea una stringa di firma con la richiesta canonica e informazioni aggiuntive, ad esempio l'algoritmo, la data della richiesta, l'ambito delle credenziali e il digest (hash) della richiesta canonica.

Fase 4: calcolo della firma

Dopo avere ottenuto la chiave di firma, puoi calcolare la firma eseguendo un'operazione hash con chiave sulla stringa di firma. Usa la chiave di firma derivata come chiave hash per questa operazione.

Fase 5: aggiunta della firma alla richiesta.

Dopo aver calcolato la firma, aggiungila a un'intestazione HTTP o alla stringa di query della richiesta.

Fase 1: creazione di una richiesta canonica

Crea una richiesta canonica concatenando le seguenti stringhe, separate da caratteri di nuova riga. Questo aiuta a garantire che la firma calcolata e la firma calcolata possano corrispondere. AWS

```
<HTTPMethod>\n
<CanonicalURI>\n
<CanonicalQueryString>\n
<CanonicalHeaders>\n
<SignedHeaders>\n
<HashedPayload>
```

- *HTTPMethod* - I metodi HTTP, come GET, PUT, HEAD, e DELETE.

- *CanonicalUri*— La versione con codifica URI dell'URI del componente del percorso assoluto, che inizia con «/» che segue il nome di dominio e fino alla fine della stringa o fino al punto interrogativo ('?') se disponi di parametri della stringa di query. Se il percorso assoluto è vuoto, usa una barra (/). L'URI nell'esempio seguente, `/examplebucket/myphoto.jpg`, è il percorso assoluto e non devi codificare "/" nel percorso assoluto:

```
http://s3.amazonaws.com/examplebucket/myphoto.jpg
```

- *CanonicalQueryString*— I parametri della stringa di query con codifica URI. Ogni nome e ogni valore vengono codificati singolarmente tramite URI. È inoltre necessario ordinare i parametri nella stringa di query canonica in ordine alfabetico in base al nome della chiave. L'ordinamento avviene dopo la codifica. La stringa di query nell'esempio di URI seguente è:

```
http://s3.amazonaws.com/examplebucket?prefix=somePrefix&marker=someMarker&max-keys=2
```

La stringa di query canonica è la seguente (le interruzioni di riga vengono aggiunte a questo esempio a fini di leggibilità):

```
UriEncode("marker")+"="+UriEncode("someMarker")+"&"+
UriEncode("max-keys")+"="+UriEncode("20") + "&" +
UriEncode("prefix")+"="+UriEncode("somePrefix")
```

Quando una richiesta ha come target una sottorisorsa, il valore del parametro di query corrispondente sarà una stringa vuota (""). Ad esempio, il seguente URI identifica la sottorisorsa `ACL` sul bucket `examplebucket`:

```
http://s3.amazonaws.com/examplebucket?acl
```

CanonicalQueryString In questo caso è il seguente:

```
UriEncode("acl") + "=" + ""
```

Se l'URI non include un '?', la richiesta non contiene una stringa di query e occorre impostare la stringa di query canonica su una stringa vuota (""). Dovrai comunque includere "\n".

- *CanonicalHeaders*— Un elenco di intestazioni di richiesta con i relativi valori. Le singole coppie di nome e valore dell'intestazione sono separate dal carattere di nuova riga ("\n"). Di seguito è riportato un esempio di canonicalheader:

```
Lowercase(<HeaderName1>)+":"+Trim(<value>)+"\n"
Lowercase(<HeaderName2>)+":"+Trim(<value>)+"\n"
...
Lowercase(<HeaderNameN>)+":"+Trim(<value>)+"\n"
```

CanonicalHeaders l'elenco deve includere quanto segue:

- Intestazione `host` HTTP.
- Se l'`Content-Type`intestazione è presente nella richiesta, è necessario aggiungerla all'*CanonicalHeaders*elenco.
- Devi aggiungere anche qualsiasi intestazione `x-amz-*` che desideri includere nella richiesta. Ad esempio, se utilizzi credenziali di sicurezza temporanee, nella tua richiesta devi includere `x-amz-security-token`. È necessario aggiungere questa intestazione nell'elenco di. *CanonicalHeaders*

> **Note**
>
> L'`x-amz-content-sha256`intestazione è necessaria per le richieste Amazon AWS S3. Fornisce un hash del payload di richiesta. Se non è presente alcun payload, devi indicare l'hash di una stringa vuota.

Il nome di ogni intestazione deve:

- usare caratteri minuscoli.
- in ordine alfabetico.
- seguiti da due punti (:).

Per i valori, devi:

- eliminare eventuali spazi all’inizio o alla fine.
- convertire gli spazi sequenziali in uno spazio singolo.
- separare i valori per un'intestazione multivalore con virgole.

- Nella firma devi includere l'intestazione dell'host (HTTP/1.1) o l'intestazione :authority (HTTP/2) e tutte le intestazioni `x-amz-*`. Facoltativamente puoi includere altre intestazioni standard nella firma, ad esempio content-type.

Le funzioni `Lowercase()` e `Trim()` utilizzate in questo esempio sono descritte nella sezione precedente.

Di seguito è riportata una stringa `CanonicalHeaders` di esempio. I nomi di intestazione sono in caratteri minuscoli e in ordine alfabetico.

```
host:s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130708T220855Z
```

> **Note**
>
> Ai fini del calcolo di una firma di autorizzazione, sono richiesti solo l'host e qualsiasi altra intestazione `x-amz-*`; tuttavia, per evitare la manomissione dei dati, nel calcolo della firma è consigliabile includere tutte le intestazioni.

- *`SignedHeaders`*— Un elenco in ordine alfabetico e separati da punto e virgola di nomi di intestazioni di richiesta in lettere minuscole. Le intestazioni della richiesta nell'elenco sono le stesse che hai incluso nella stringa `CanonicalHeaders`. Ad esempio, per l'esempio precedente, il valore di sarebbe il seguente: *`SignedHeaders`*

```
host;x-amz-content-sha256;x-amz-date
```

- *`HashedPayload`*— Una stringa creata utilizzando il payload nel corpo della richiesta HTTP come input per una funzione hash. Questa stringa utilizza caratteri esadecimali minuscoli.

```
Hex(SHA256Hash(<payload>)
```

Se non è presente alcun payload nella richiesta, calcola un hash della stringa vuota come segue:

```
Hex(SHA256Hash(""))
```

L'hash restituisce i seguenti valori:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Ad esempio, quando carichi un oggetto utilizzando una richiesta PUT, fornisci i dati dell'oggetto nel corpo. Quando recuperi un oggetto utilizzando una richiesta GET, calcoli l'hash della stringa vuota.

Fase 2: creazione di un hash della richiesta canonica

Crea un hash (digest) della richiesta canonica con lo stesso algoritmo utilizzato per creare l'hash del payload. L'hash della richiesta canonica è una stringa di caratteri esadecimali minuscoli.

Fase 3: creazione di una stringa da firmare

Crea una richiesta concatenando le seguenti stringhe, separate da caratteri di nuova riga. Non terminare questa stringa con un carattere di nuova riga.

```
Algorithm \n
RequestDateTime \n
CredentialScope  \n
HashedCanonicalRequest
```

- *Algorithm*: l'algoritmo utilizzato per creare l'hash della richiesta canonica. Per SHA-256, l'algoritmo è AWS4-HMAC-SHA256.
- *RequestDateTime*— La data e l'ora utilizzate nell'ambito delle credenziali. Questo valore è l'ora UTC corrente in formato ISO 8601 (ad esempio,20130524T000000Z).
- *CredentialScope*— L'ambito delle credenziali. Ciò limita la firma risultante alla regione e al servizio specificati. La stringa ha il seguente formato: *AAAAMMGG*/*regione*/*servizio*/aws4_request.
- *HashedCanonicalRequest*— L'hash della richiesta canonica. Questo valore viene calcolato nella fase 2.

Di seguito è riportata una stringa di esempio da firmare.

```
"AWS4-HMAC-SHA256" + "\n" +
timeStampISO8601Format + "\n" +
```

```
<Scope> + "\n" +
Hex(SHA256Hash(<CanonicalRequest>))
```

Fase 4: calcolo della firma

Nella versione 4 di AWS Signature, invece di utilizzare le chiavi di AWS accesso per firmare una richiesta, crei una chiave di firma valida per una regione e un servizio specifici come informazioni di autenticazione da aggiungere alla richiesta.

```
DateKey = HMAC-SHA256("AWS4"+"<SecretAccessKey>", "<YYYYMMDD>")
DateRegionKey = HMAC-SHA256(<DateKey>, "<aws-region>")
DateRegionServiceKey = HMAC-SHA256(<DateRegionKey>, "<aws-service>")
SigningKey = HMAC-SHA256(<DateRegionServiceKey>, "aws4_request")
```

Per un elenco dei codici delle regioni, consulta la pagina Endpoint regionali nei AWS Riferimenti generali .

Per ogni passaggio, richiama la funzione hash con la chiave e i dati richiesti. Il risultato di ogni chiamata alla funzione hash diventa l'input per la chiamata successiva alla funzione.

Input richiesto

- Una stringa, `Key`, che contiene la tua chiave di accesso segreta
- Una stringa, `Date`, che contiene la data utilizzata nell'ambito delle credenziali, nel formato AAAAMMGG
- Una stringa, `Region`, che contiene il codice della regione (ad esempio, `us-east-1`)
- Una stringa, `Service`, che contiene il codice del servizio (ad esempio, `ec2`)
- La stringa da firmare creata nel passaggio precedente.

Calcolo della firma

1. Concatena "AWS4" e la chiave di accesso segreta. Chiama la funzione hash con la stringa concatenata come stringa di chiave e data come dati.

   ```
   kDate = hash("AWS4" + Key, Date)
   ```

2. Chiama la funzione hash con il risultato della chiamata precedente come stringa di chiave e regione come dati.

```
kRegion = hash(kDate, Region)
```

3. Chiama la funzione hash con il risultato della chiamata precedente come stringa di chiave e servizio come dati.

```
kService = hash(kRegion, Service)
```

4. Chiama la funzione hash con il risultato della chiamata precedente come chiave e "aws4_request" come dati.

```
kSigning = hash(kService, "aws4_request")
```

5. Chiama la funzione hash con il risultato della chiamata precedente come chiave e stringa da firmare come dati. Il risultato è la firma come valore binario.

```
signature = hash(kSigning, string-to-sign)
```

6. Converti la firma da rappresentazione binaria a esadecimale, in caratteri minuscoli.

Fase 5: aggiunta della firma alla richiesta.

Example Esempio: intestazione di autorizzazione

Nell'esempio seguente viene mostrata una intestazione `Authorization` per l'operazione `DescribeInstances`. Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga. Nel tuo codice, deve essere una stringa continua. Non vi è alcun virgola tra l'algoritmo e `Credential`. Tuttavia, gli altri elementi devono essere separati da virgole.

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20220830/us-east-1/ec2/aws4_request,
SignedHeaders=host;x-amz-date,
Signature=calculated-signature
```

Example Esempio: richiesta con parametri di autenticazione nella stringa di query

L'esempio seguente mostra una query per l'operazione `DescribeInstances` che include le informazioni di autenticazione. Per motivi di leggibilità, questo esempio è formattato con interruzioni di riga e non è codificato con l'URL. Nel codice, la stringa di query deve essere una stringa continua con codifica URL.

```
https://ec2.amazonaws.com/?
Action=DescribeInstances&
Version=2016-11-15&
X-Amz-Algorithm=AWS4-HMAC-SHA256&
X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20220830/us-east-1/ec2/aws4_request&
X-Amz-Date=20220830T123600Z&
X-Amz-SignedHeaders=host;x-amz-date&
X-Amz-Signature=calculated-signature
```

Codice sorgente negli SDK AWS

Gli AWS SDK includono il codice sorgente GitHub per la firma delle richieste AWS API. Per alcuni esempi di codice, consulta Progetti di esempio nell'archivio di AWS esempi

- AWS SDK for .NET — AWS4Signer.cs
- AWS SDK for C++ — AWSAuthV4Signer.cpp
- AWS SDK for Go — v4.go
- AWS SDK for Java — 4Signer.java BaseAws
- AWS SDK for JavaScript — v4.js
- AWS SDK for PHP — SignatureV4.php
- AWS SDK for Python (Boto) — signers.py
- AWS SDK for Ruby — signer.rb

Richiesta di esempi di firma

I seguenti esempi di richieste di AWS firma mostrano come utilizzare SigV4 per firmare le richieste inviate senza l' AWS SDK o lo strumento da riga di AWS comando.

Caricamento di Amazon S3 basato su browser tramite HTTP POST

Richieste di autenticazione: caricamenti basati su browser descrive la firma e le informazioni pertinenti che Amazon S3 utilizza per calcolare la firma al ricevimento della richiesta.

Esempio: il caricamento basato su browser tramite HTTP POST (utilizzando la versione 4 di AWS Signature) fornisce ulteriori informazioni con un esempio di policy POST e un modulo che è possibile utilizzare per caricare un file. La policy di esempio e le credenziali fittizie mostrano il flusso di lavoro e la firma e l'hash della policy risultanti.

Richieste autenticate VPC Lattice

Esempi di richieste autenticate Signature Version 4 (SigV4) fornisce esempi in Python e Java che mostrano come eseguire la firma delle richieste con e senza intercettori personalizzati.

Utilizzo di Signature Version 4 con Amazon Translate

Utilizzo di Signature Version 4 con Amazon Translate mostra come utilizzare un programma Python per aggiungere informazioni di autenticazione alle richieste di Amazon Translate. L'esempio effettua una richiesta POST, crea una struttura JSON contenente il testo da tradurre nel corpo (payload) della richiesta e trasferisce le informazioni di autenticazione in un'intestazione di autorizzazione.

Utilizzo di Signature Version 4 con Neptune

Esempio: connessione a Neptune utilizzando Python con firma Signature Version 4 mostra come effettuare richieste firmate a Neptune usando Python. Questo esempio include varianti per l'utilizzo di una chiave di accesso o di credenziali temporanee.

Firma delle richieste HTTP a S3 Glacier

Esempio di calcolo della firma per l'API di streaming illustra i dettagli della creazione di una firma per il caricamento dell'archivio (POST), una delle due API di streaming di S3 Glacier.

Invio di richieste HTTP ad Amazon SWF

Invio di richieste HTTP ad Amazon SWF mostra il contenuto dell'intestazione per una richiesta JSON ad Amazon SWF.

Calcolo delle firme per le API di streaming in Amazon Service OpenSearch

La firma di una richiesta di ricerca Amazon OpenSearch Service con AWS SDK for PHP versione 3 include un esempio di come inviare richieste HTTP firmate ad OpenSearch Amazon Service.

Progetti di esempio nell'archivio di AWS esempi

I seguenti progetti di esempio mostrano come firmare le richieste per effettuare richieste API Rest a AWS servizi con linguaggi comuni come Python, Node.js, Java, C#, Go e Rust.

Progetti Signature Version 4a

Il progetto sigv4-signing-examples fornisce esempi di come firmare le richieste con SigV4A per effettuare richieste API Rest con linguaggi Servizi AWS comuni come Python, Node.js, Java, C#, Go e Rust.

Il a-signing-examples progetto sigv4 fornisce esempi per firmare richieste API multiregionali, ad esempio punti di accesso multiregionali in Amazon S3.

Pubblica su AWS IoT Core

Il codice Python da pubblicare AWS IoT Core utilizzando il protocollo HTTPS fornisce indicazioni su come pubblicare messaggi AWS IoT Core utilizzando il protocollo HTTPS e l'autenticazione AWS SigV4. Ha due implementazioni di riferimento: una in Python e l'altra in. NodeJs

L'applicazione.Net Framework su cui pubblicare AWS IoT Core utilizzando il protocollo HTTPS fornisce indicazioni su come pubblicare messaggi AWS IoT Core utilizzando il protocollo HTTPS e l'autenticazione AWS SigV4. Questo progetto include anche un'implementazione equivalente a .NET core.

Risolvi i problemi relativi alle richieste firmate per le API AWS

> **Important**
>
> A meno che non utilizzi gli AWS SDK o la CLI, devi scrivere codice per calcolare le firme che forniscono informazioni di autenticazione nelle tue richieste. Il calcolo della firma SigV4 può essere un'operazione complessa e consigliamo di utilizzare gli SDK o la CLI AWS quando possibile.

Quando sviluppi codice che crea una richiesta firmata, potresti ricevere HTTP 403 da. `SignatureDoesNotMatch` Servizi AWS Questi errori indicano che il valore della firma nella richiesta HTTP AWS non corrisponde alla firma Servizio AWS calcolata. Gli errori HTTP 401 `Unauthorized` vengono restituiti quando le autorizzazioni non consentono al chiamante di effettuare la richiesta.

Le richieste API potrebbero restituire un errore se:

- La richiesta API non è firmata e utilizza l'autenticazione IAM.

- Le credenziali IAM utilizzate per firmare la richiesta non sono corrette o non dispongono delle autorizzazioni per richiamare l'API.
- La firma della richiesta API firmata non corrisponde alla firma calcolata dal servizio AWS .
- L'intestazione della richiesta API non è corretta.

> **Note**
>
> Aggiorna il protocollo di AWS firma da Signature versione 2 (SigV2) a AWS Signature versione 4 (SigV4) prima di esplorare altre soluzioni di errore. Servizi come Amazon S3 e le regioni non supportano più la firma SigV2.

Possibili cause

- Errori delle credenziali
- Errori nella richiesta canonica e nella stringa di firma
- Errori nell'ambito delle credenziali
- Errori nella chiave di firma

Errori delle credenziali

Assicurati che la richiesta dell'API sia firmata con SigV4. Se la richiesta API non è firmata, potresti ricevere l'errore: `Missing Authentication Token`. Aggiungi la firma mancante e invia nuovamente la richiesta.

Verifica che le credenziali di autenticazione della chiave di accesso e della chiave segreta siano corrette. Se la chiave di accesso non è corretta, potresti ricevere l'errore: `Unauthorized`. Assicurati che l'entità utilizzata per firmare la richiesta sia autorizzata a effettuare la richiesta. Per informazioni dettagliate, vedi Risoluzione dei problemi dei messaggi di errore di accesso rifiutato.

Errori nella richiesta canonica e nella stringa di firma

Se hai calcolato la richiesta canonica in Fase 2: creazione di un hash della richiesta canonica o Fase 3: creazione di una stringa da firmare, la fase di verifica della firma eseguita dal servizio ha esito negativo con li seguente messaggio di errore:

```
The request signature we calculated does not match the signature you provided
```

Quando il AWS servizio riceve una richiesta firmata, ricalcola la firma. Se sussistono differenze nei valori, le firme non corrispondono. Confronta la stringa e la richiesta canonica con la tua richiesta firmata con il valore nel messaggio di errore. Modifica il processo di firma se riscontri differenze.

> **Note**
>
> Puoi anche verificare di non aver inviato la richiesta tramite una proxy che modifica le intestazioni o la richiesta.

Example Esempio di richiesta canonica

```
GET                                                    -------- HTTP method
/                                                      -------- Path. For API stage
 endpoint, it should be /{stage-name}/{resource-path}
                                                       -------- Query string key-
value pair. Leave it blank if the request doesn't have a query string.
content-type:application/json                          -------- Header key-value
 pair. One header per line.
host:0123456789.execute-api.us-east-1.amazonaws.com    -------- Host and x-amz-date
 are required headers for all signed requests.
x-amz-date:20220806T024003Z

content-type;host;x-amz-date                           -------- A list of signed
 headers
d167e99c53f15b0c105101d468ae35a3dc9187839ca081095e340f3649a04501       -------- Hash
 of the payload
```

Per verificare che la chiave segreta corrisponda all'ID della chiave di accesso, puoi testarla con un'implementazione funzionante nota. Ad esempio, utilizza un AWS SDK o la AWS CLI per effettuare una richiesta a. AWS

Intestazione della richiesta API

Assicurati che l'intestazione di autorizzazione SigV4 che hai aggiunto in Fase 4: calcolo della firma includa la chiave di credenziale corretta, simile alla seguente:

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
```

```
Signature=example-generated-signature
```

Se la chiave di credenziale è mancante o errata, potresti ricevere l'errore: `Authorization header requires 'Credential' parameter. Authorization header requires 'Signature' parameter.` Assicurati che la richiesta di autorizzazione SigV4 includa anche la data della richiesta utilizzando `HTTP Date` o l'intestazione `x-amz-date`.

Errori nell'ambito delle credenziali

L'ambito delle credenziali che hai creato in Fase 3: creazione di una stringa da firmare limita una firma a una data, una regione e un servizio specifici. Questa stringa ha il seguente formato:

```
YYYYMMDD/region/service/aws4_request
```

> **Note**
>
> Se si utilizza SigV4a, la regione non è inclusa nell'ambito delle credenziali.

Data

Se l'ambito delle credenziali non specifica la stessa data dell'intestazione di x-amz-date, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Date in Credential scope does not match YYYYMMDD from ISO-8601 version of date from
 HTTP
```

Se la richiesta specifica un orario futuro, la fase di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Signature not yet current: date is still later than date
```

Se la richiesta è scaduta, la fase di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Signature expired: date is now earlier than date
```

Regione

Se l'ambito delle credenziali non specifica la stessa regione della richiesta, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped to a valid Region, not region-code
```

Servizio

Se l'ambito delle credenziali non specifica lo stesso servizio dell'intestazione di host, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped to correct service: 'service'
```

Stringa di terminazione

Se l'ambito delle credenziali termina con aws4_request, il passaggio di verifica della firma fallisce e viene visualizzato il seguente messaggio di errore:

```
Credential should be scoped with a valid terminator: 'aws4_request'
```

Errori nella chiave di firma

Gli errori causati da un'errata derivazione della chiave di firma o dall'uso improprio della crittografia sono più difficili da risolvere. Dopo aver verificato che la stringa canonica e la stringa da firmare siano corrette, puoi anche verificare la presenza di uno dei seguenti problemi:

- La chiave di accesso segreta non corrisponde all'ID della chiave di accesso specificato.
- Si è verificato un problema con il codice di derivazione della chiave.

Per verificare che la chiave segreta corrisponda all'ID della chiave di accesso, puoi testarla con un'implementazione funzionante nota. Ad esempio, usa un AWS SDK o il AWS CLI per effettuare una richiesta a. AWS Per alcuni esempi, consultare Richiesta di esempi di firma.

Riferimento alla policy JSON IAM

Questa sezione presenta la sintassi, le descrizioni e gli esempi dettagliati di elementi, variabili e logica di valutazione delle policy JSON in IAM. Per ulteriori informazioni generali, consulta Panoramica delle policy JSON.

Questo riferimento include le seguenti sezioni.

- Documentazione di riferimento degli elementi delle policy JSON IAM: ulteriori informazioni sugli elementi che è possibile utilizzare durante la creazione di una policy. Visualizzare ulteriori esempi di policy e ulteriori informazioni su condizioni, tipi di dati supportati e il modo in cui vengono utilizzati in vari servizi.
- Logica di valutazione delle policy— Questa sezione descrive AWS le richieste, come vengono autenticate e come vengono AWS utilizzate le politiche per determinare l'accesso alle risorse.
- Sintassi del linguaggio della policy JSON IAM : questa sezione presenta una sintassi formale per il linguaggio utilizzato per creare le policy in IAM.
- AWS politiche gestite per le funzioni lavorative: questa sezione elenca tutte le policy gestite di AWS che mappano direttamente a funzioni lavorative nel settore IT. Utilizzare queste policy per concedere le autorizzazioni necessarie per eseguire le attività che ci si aspetta da qualcuno in una determinata funzione lavorativa. Queste policy consolidano le autorizzazioni per molti servizi in una singola policy.
- AWS chiavi di contesto della condizione globale— Questa sezione include un elenco di tutte le chiavi di condizione AWS globali che è possibile utilizzare per limitare le autorizzazioni in una policy IAM.
- chiavi contestuali IAM e AWS STS condition— Questa sezione include un elenco di tutte le chiavi IAM e di AWS STS condizione che è possibile utilizzare per limitare le autorizzazioni in una policy IAM.
- Azioni, risorse e chiavi di condizione per AWS i servizi: questa sezione presenta un elenco di tutte le operazioni AWS API che è possibile utilizzare come autorizzazioni in una policy IAM. Include anche le chiavi di condizione specifiche del servizio che possono essere utilizzate per ottimizzare ulteriormente la richiesta.

Documentazione di riferimento degli elementi delle policy JSON IAM

I documenti delle policy JSON sono costituiti da elementi. Gli elementi vengono elencati qui nell'ordine generale in cui vengono utilizzati in una policy. L'ordine degli elementi non ha importanza, ad esempio l'elemento `Resource` può venire prima dell'elemento `Action`. Non devi specificare alcun elemento `Condition` nella policy. Per ulteriori informazioni sulla struttura generale e lo scopo di un documento di policy JSON, consulta la pagina Panoramica delle policy JSON.

Alcuni elementi della policy JSON sono reciprocamente esclusivi. Questo significa che non puoi creare una policy che utilizza entrambi. Ad esempio, non è possibile utilizzare `Action` e

`NotAction` nella stessa dichiarazione di policy. Altre coppie che si escludono reciprocamente sono `Principal`/`NotPrincipal` e `Resource`/`NotResource`.

I dettagli di ciò che va a comporre una policy variano per ciascun servizio, a seconda di quali operazioni il servizio rende disponibili, quali tipi di risorse contiene e così via. Quando stai scrivendo delle policy per un servizio specifico, è utile consultare esempi di policy per quel servizio. Per un elenco di tutti i servizi che supportano IAM e per i collegamenti alla documentazione di quei servizi che illustrano IAM e le policy, consulta AWS servizi che funzionano con IAM.

Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

Argomenti

Elementi delle policy JSON IAM: Version

> **Chiarimento**
>
> Questo elemento della policy JSON `Version` è diverso da una versione della policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per informazioni sul supporto per versioni multiple disponibile per le policy gestite, consultare the section called “Controllo delle versioni delle policy IAM”.

L'elemento della policy `Version` specifica le regole sintattiche di linguaggio che devono essere utilizzate per elaborare una policy. Per utilizzare tutte le funzionalità disponibili della policy, includi il seguente elemento `Version` all'esterno dell'elemento `Statement` in tutte le policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

IAM supporta i seguenti valori degli elementi `Version`:

- `2012-10-17`. Questa è la versione corrente del linguaggio della policy e deve sempre includere un elemento `Version` ed essere impostato su `2012-10-17`. In caso contrario, non è possibile utilizzare caratteristiche come variabili di policy introdotte con questa versione.
- `2008-10-17`. Questa è una versione precedente del linguaggio della policy. Potresti vedere questa versione su policy esistenti meno recenti. Non utilizzare questa versione per le nuove policy o quando si aggiornano policy esistenti. Le caratteristiche più recenti, come variabili di policy, non funzioneranno con la tua policy. Ad esempio, le variabili tipo `${aws:username}` non saranno riconosciute come variabili e verranno trattate come stringhe letterali nella policy.

Elementi delle policy JSON IAM: Id

L'elemento `Id` specifica un identificatore opzionale per la policy. L'ID viene utilizzato in modo diverso in servizi diversi. L'ID è consentito nelle policy basate su risorse, ma non nelle policy basate sulle identità.

Per i servizi che consentono di impostare un elemento `ID`, consigliamo di utilizzare un UUID (GUID) per il valore o incorporare un UUID come parte dell'ID per garantire l'univocità.

```
{
  "Version": "2012-10-17",
  "Id": "cd3ad3d9-2776-4ef1-a904-4c229d1642ee",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Alcuni AWS servizi (ad esempio Amazon SQS o Amazon SNS) potrebbero richiedere questo elemento e avere requisiti di unicità per esso. Per informazioni specifiche per servizio sulla scrittura di policy, consultare la documentazione per il servizio in uso.

Elementi delle policy JSON IAM: Statement

L'elemento `Statement` è l'elemento principale per una policy. Questa elemento è obbligatorio. L'elemento `Statement` può contenere una singola istruzione o una matrice di singole istruzioni. Ogni singolo blocco di istruzioni deve essere racchiuso tra parentesi graffe { }. In caso di istruzioni multiple, l'array deve essere racchiuso tra parentesi quadre [ ].

```
"Statement": [{...},{...},{...}]
```

L'esempio seguente mostra una policy che contiene una serie di tre istruzioni all'interno di un singolo elemento `Statement`. La policy consente di accedere alla propria "cartella home" nella console

Amazon S3. La policy include la variabile `aws:username`, che viene sostituita durante la valutazione della policy con il nome utente dalla richiesta. Per ulteriori informazioni, consulta Introduzione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::BUCKET-NAME",
      "Condition": {"StringLike": {"s3:prefix": [
        "",
        "home/",
        "home/${aws:username}/"
      ]}}
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}",
        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}/*"
      ]
    }
  ]
}
```

Elementi delle policy JSON IAM: Sid

È possibile fornire un `Sid` (ID della dichiarazione) come identificatore opzionale per l'informativa sulla politica. Puoi assegnare un valore `Sid` a ogni istruzione in una matrice di istruzioni. È possibile utilizzare il valore `Sid` come descrizione per l'istruzione della policy. In servizi che consentono di specificare un elemento `ID`, ad esempio SQS e SNS, il valore `Sid` è semplicemente un ID

secondario dell'ID del documento di policy. In IAM, il valore `Sid` deve essere univoco all'interno di una policy JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStatementID",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

L'elemento `Sid` supporta lettere maiuscole ASCII (A-Z), lettere minuscole (a-z) e numeri (0-9).

IAM non utilizza il `Sid` nell'API IAM. Non puoi recuperare una determinata istruzione in base a questo ID.

> **Note**
>
> Alcuni AWS servizi (ad esempio Amazon SQS o Amazon SNS) potrebbero richiedere questo elemento e avere requisiti di unicità per esso. Per informazioni specifiche del servizio sulla scrittura di policy, consulta la documentazione per il servizio in uso.

Elementi delle policy JSON IAM: Effect

L'elemento `Effect` è obbligatorio e specifica se l'istruzione determina un consenso o un rifiuto esplicito. I valori validi di `Effect` sono `Allow` e `Deny`. Il valore `Effect` prevede la distinzione tra lettere maiuscole e minuscole.

```
"Effect":"Allow"
```

Come impostazione predefinita, l'accesso alle risorse è negato. Per consentire l'accesso a una risorsa, è necessario impostare l'elemento `Effect` su `Allow`. Per ignorare un consenso (ad esempio, per ignorare un consenso altrimenti valido), è necessario impostare l'elemento `Effect` su `Deny`. Per ulteriori informazioni, consulta Logica di valutazione delle policy.

AWS Elementi della policy JSON: Principal

Utilizzare l'elemento `Principal` in una policy JSON basata sulle risorse per specificare il principale a cui è consentito o negato l'accesso a una risorsa.

Nelle policy basate sulle risorse devi utilizzare l'elemento `Principal`. Diversi servizi supportano le policy basate sulle risorse, tra cui IAM. Il tipo di policy basata sulle risorse IAM è una policy di attendibilità del ruolo. Nei ruoli IAM, utilizza l'elemento `Principal` nella policy di attendibilità del ruolo per specificare chi può assumere il ruolo. Per l'accesso tra account, è necessario specificare l'identificatore a 12 cifre dell'account affidabile. Per capire se i principali negli account esterni alla zona di attendibilità (organizzazione o account attendibile) dispongono dell'accesso per assumere i ruoli, consulta Cos'è IAM Access Analyzer?.

> **Note**
>
> Dopo aver creato il ruolo, è possibile modificare l'account in "*" per consentire a tutti di assumere il ruolo. In questo caso, è consigliabile limitare gli utenti che possono accedere al ruolo attraverso altri mezzi, ad esempio un elemento `Condition` che limita l'accesso solo a determinati indirizzi IP. Non permettere che il ruolo sia accessibile a tutti.

Altri esempi di risorse che supportano le policy basate sulle risorse includono un bucket Amazon S3 o AWS KMS key.

Non puoi usare l'elemento `Principal` in una policy basata su identità Le policy basate su identità sono policy di autorizzazione che si collegano a identità IAM (utenti, gruppi o ruoli). In questi casi, il principale è implicito nell'identità dove è collegata la policy.

Argomenti

- Specifica di un'entità principale
- Account AWS presidi
- Principali ruolo IAM
- Principali della sessione come ruolo
- Principali dell'utente IAM
- Principi fondamentali di Centro identità IAM
- AWS STS principi di sessione utente federati
- AWS presidi del servizio

- AWS i principali del servizio nelle regioni che accettano l'adesione
- Tutti i principali
- Ulteriori informazioni

Specifica di un'entità principale

È possibile specificare un principale nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali.

In una policy è possibile specificare una delle seguenti entità:

- Account AWS e utente root
- Ruoli IAM
- Sessioni come ruolo
- Utenti IAM
- Sessioni come utente federato
- AWS servizi
- Tutti i principali

Non è possibile identificare un gruppo di utenti come principale in una policy (ad esempio una policy basata sulle risorse) perché i gruppi si riferiscono alle autorizzazioni, non all'autenticazione, e i principali sono entità IAM autenticate.

È possibile specificare più di un principale per ciascuno dei tipi di entità nelle sezioni seguenti utilizzando un array. Gli array possono richiedere uno o più valori. Quando si specifica più di un principale in un elemento, si concedono le autorizzazioni a ciascun principale. Questo è un `OR` logico e non un `AND` logico, perché si viene autenticati come un principale alla volta. Se includi più di un valore, utilizza parentesi quadre (`[` e `]`) e delimita con le virgole ogni voce per l'array. La seguente policy di esempio definisce le autorizzazioni per l'account 123456789012 o per l'account 555555555555.

```
"Principal" : {
"AWS": [
  "123456789012",
  "555555555555"
  ]
```

```
}
```

> **Note**
>
> Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN.

Account AWS presidi

È possibile specificare Account AWS gli identificatori nell'`Principal`elemento di una politica basata sulle risorse o nelle chiavi di condizione che supportano i principali. In questo modo l'autorità viene delegata all'account. Quando consenti l'accesso a un altro account, un amministratore di tale account deve concedere l'accesso a un'identità (utente o ruolo IAM) in tale account. Quando si specifica un Account AWS, è possibile utilizzare l'account ARN (arn:aws:iam:: *`account-ID:root`*) o un modulo abbreviato costituito dal prefisso seguito dall'ID dell'account. `"AWS":`

Ad esempio, fornendo un account ID di `123456789012`, è possibile utilizzare uno dei seguenti metodi per specificare l'account nell'elemento `Principal`:

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

```
"Principal": { "AWS": "123456789012" }
```

L'ARN dell'account e l'ID dell'account abbreviato si comportano allo stesso modo: entrambi delegano le autorizzazioni all'account. L'utilizzo dell'ARN dell'account nell'elemento `Principal` non limita le autorizzazioni solo per l'utente root dell'account.

> **Note**
>
> Quando salvi una policy basata sulle risorse che include l'ID dell'account abbreviato, il servizio potrebbe convertirlo nell'ARN del principale. Ciò non modifica la funzionalità della policy.

Alcuni servizi supportano opzioni aggiuntive per specificare l'intestatario dell'account. AWS Ad esempio, Amazon S3 ti consente di specificare un ID utente canonico utilizzando il formato seguente:

```
"Principal": { "CanonicalUser":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be" }
```

È inoltre possibile specificarne più di uno Account AWS(o un ID utente canonico) come principale utilizzando un array. Ad esempio, è possibile specificare un principale in una policy del bucket utilizzando tutti e tre i metodi.

```
"Principal": {
  "AWS": [
    "arn:aws:iam::123456789012:root",
    "999999999999"
  ],
  "CanonicalUser": "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
}
```

Principali ruolo IAM

È possibile specificare gli ARN dei principali del ruolo IAM nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali. I ruoli IAM sono identità. In IAM, le identità sono risorse a cui è possibile assegnare autorizzazioni. I ruoli si affidano a un'altra identità autenticata per assumere tale ruolo. Ciò include un principale in AWS o un utente di un provider di identità esterno (IdP). Quando un principal o un'identità assume un ruolo, ricevono credenziali di sicurezza temporanee con le autorizzazioni del ruolo assunto. Quando utilizzano tali credenziali di sessione per eseguire operazioni in AWS, diventano i principali della sessione di ruolo.

I ruoli IAM sono identità esistenti in IAM. I ruoli si affidano a un'altra identità autenticata, ad esempio un titolare AWS o un utente di un provider di identità esterno. Quando un principal o un'identità assume un ruolo, ricevono credenziali di sicurezza temporanee. Possono quindi utilizzare tali credenziali come principale della sessione come ruolo per eseguire operazioni in AWS.

Quando si specifica un principale del ruolo in una policy basata sulle risorse, le autorizzazioni effettive per il principale sono limitate da qualsiasi tipo di policy che limita le autorizzazioni per il ruolo. Ciò include le policy di sessione e i limiti delle autorizzazioni. Per ulteriori informazioni su come vengono valutate le autorizzazioni effettive per una sessione come ruolo, consulta Logica di valutazione delle policy.

Per specificare l'ARN del ruolo nell'elemento `Principal`, utilizza questo formato:

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:role/role-name" }
```

> **Important**
>
> Se l'elemento `Principal` in una policy di attendibilità del ruolo contiene un ARN che punta a un determinato ruolo IAM, allora l'ARN si trasforma nell'ID principale univoco del ruolo quando si salva la policy. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo. Questa ID nella console non è normalmente presente, in quanto IAM usa una trasformazione inversa verso l'ARN del ruolo quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina il ruolo, la relazione viene interrotta. La policy non è più applicabile, anche se si ricrea il ruolo perché il nuovo ruolo ha un nuovo ID principale che non corrisponde all'ID principale archiviato nella policy di affidabilità. Quando ciò accade, l'ID principale viene visualizzato nelle politiche basate sulle risorse perché non è più AWS possibile mapparlo su un ARN valido. Il risultato finale è che se si elimina e si ricrea un ruolo referenziato in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo nella policy per sostituire l'ID principale con il nome ARN corretto. L'ARN si trasforma nuovamente nel nuovo ID principale del ruolo quando si salva la policy.

In alternativa, è possibile specificare il principale del ruolo come principale in una policy basata sulle risorse oppure creare una policy di ampia autorizzazione che usa la chiave di condizione `aws:PrincipalArn`. Quando si utilizza questa chiave, al principale della sessione come ruolo vengono concesse le autorizzazioni in base all'ARN del ruolo assunto e non all'ARN della sessione risultante. Poiché AWS non converte gli ARN delle chiavi di condizione in ID, le autorizzazioni concesse al ruolo ARN persistono se si elimina il ruolo e quindi si crea un nuovo ruolo con lo stesso nome. I tipi di policy basati su identità, come i limiti delle autorizzazioni o le policy di sessione, non limitano le autorizzazioni concesse tramite la chiave di condizione `aws:PrincipalArn` con un carattere jolly (*) nell'elemento `Principal`, a meno che le policy basate su identità non contengano un rifiuto esplicito.

Principali della sessione come ruolo

È possibile specificare le sessioni come ruolo nell'elemento `Principal` di una policy basata sulle risorse o in chiavi in condizione che supportano i principali. Quando un principal o un'identità assume un ruolo, ricevono credenziali di sicurezza temporanee con le autorizzazioni del ruolo assunto. Quando utilizzano tali credenziali di sessione per eseguire operazioni AWS, diventano responsabili della sessione di ruolo.

Il formato utilizzato per un responsabile della sessione di ruolo dipende dall' AWS STS operazione utilizzata per assumere il ruolo.

Inoltre, gli amministratori possono progettare un processo per controllare il modo di emissione delle sessioni come ruolo. Ad esempio, possono fornire una soluzione con un clic per gli utenti che creano un nome della sessione prevedibile. Se l'amministratore compie questa operazione, è possibile utilizzare i principali della sessione come ruolo nelle policy o nelle chiavi di condizione. In caso contrario, è possibile specificare l'ARN del ruolo come principale nella chiave di condizione `aws:PrincipalArn`. Il modo in cui si specifica il ruolo come principale può modificare le autorizzazioni effettive per la sessione risultante. Per ulteriori informazioni, consulta Principali ruolo IAM.

Principali di sessione del ruolo assunto

Un principale di sessione con ruolo presunto è un principale di sessione che risulta dall'utilizzo dell'operazione. AWS STS `AssumeRole` Per ulteriori informazioni su quali principali possono assumere un ruolo utilizzando questa operazione, consulta Confronto delle operazioni AWS STS API.

Per specificare l'ARN della sessione come ruolo assunto nell'elemento `Principal`, utilizza questo formato:

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:assumed-role/role-name/role-session-name" }
```

Quando si specifica una sessione con assunzione di ruolo in un elemento `Principal`, non è possibile utilizzare un carattere jolly "*" per indicare tutte le sessioni. Le entità devono sempre fare riferimento a una sessione specifica.

Principi della sessione OIDC

Un principale di sessione OIDC è un principale di sessione che risulta dall'utilizzo dell'operazione. AWS STS `AssumeRoleWithWebIdentity` Puoi utilizzare un provider OIDC (IdP) esterno per accedere e quindi assumere un ruolo IAM utilizzando questa operazione. Ciò sfrutta la federazione delle identità e genera una sessione come ruolo. Per ulteriori informazioni su quali principali possono assumere un ruolo utilizzando questa operazione, consulta Confronto delle operazioni AWS STS API.

Quando si assegna un ruolo da un provider OIDC, si ottiene questo tipo speciale di sessione principale che include informazioni sul provider OIDC.

Utilizzare questo tipo di principale nella policy per consentire o negare l'accesso in base al provider di identità Web attendibile. Per specificare l'ARN della sessione di ruolo OIDC nell'`Principal`elemento di una policy di attendibilità dei ruoli, utilizzare il formato seguente:

```
"Principal": { "Federated": "cognito-identity.amazonaws.com" }
```

```
"Principal": { "Federated": "www.amazon.com" }
```

```
"Principal": { "Federated": "graph.facebook.com" }
```

```
"Principal": { "Federated": "accounts.google.com" }
```

Principali della sessione SAML

Un principale di sessione SAML è un principale di sessione che risulta dall'utilizzo dell'operazione. AWS STS `AssumeRoleWithSAML` È possibile utilizzare un provider di identità SAML (IdP) esterno per accedere e quindi assumere un ruolo IAM utilizzando questa operazione. Ciò sfrutta la federazione delle identità e genera una sessione come ruolo. Per ulteriori informazioni su quali principali possono assumere un ruolo utilizzando questa operazione, consulta Confronto delle operazioni AWS STS API.

Quando si emette un ruolo da un provider di identità SAML, si ottiene questo tipo speciale di principale di sessione che include informazioni sul provider di identità SAML.

Utilizza questo tipo di principale nella policy per consentire o negare l'accesso in base al provider di identità SAML attendibile. Per specificare l'ARN della sessione del ruolo dell'identità Web nell'elemento `Principal` di una policy di attendibilità dei ruoli, utilizza questo formato:

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-ID:saml-provider/provider-name" }
```

Principali dell'utente IAM

Puoi specificare gli utenti IAM nell'elemento `Principal` di una policy basata sulle risorse o nelle chiavi della condizione che supportano i principali.

> **Note**
>
> In un elemento `Principal`, la parte del nome utente dell'Amazon Resource Name(ARN) fa distinzione tra maiuscole e minuscole.

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:user/user-name" }
```

```
"Principal": {
  "AWS": [
    "arn:aws:iam::AWS-account-ID:user/user-name-1",
    "arn:aws:iam::AWS-account-ID:user/user-name-2"
  ]
}
```

Quando si specificano gli utenti in un elemento `Principal`, non è possibile utilizzare un carattere jolly (`*`) che indica "tutti gli utenti". I principali devono sempre nominare utenti determinati.

> **Important**
>
> Se l'elemento `Principal` in una policy di attendibilità del ruolo contiene un nome ARN che punta a un determinato utente o IAM, allora IAM trasforma l'ARN nell'ID principale univoco dell'utente quando la policy viene salvata. Ciò aiuta a mitigare il rischio che qualcuno aumenti i propri privilegi rimuovendo e ricreando il ruolo o l'utente. Questa ID nella console non è normalmente presente, in quanto c'è anche una trasformazione inversa verso il nome ARN dell'utente quando la policy di affidabilità viene visualizzata. Tuttavia, se si elimina l'utente, la relazione viene interrotta. La policy non è più applicabile, anche se viene ricreato l'utente. Questo perché il nuovo utente ha un nuovo ID principale che non corrisponde all'ID archiviato nella policy di affidabilità. Quando ciò accade, l'ID principale viene visualizzato nelle politiche basate sulle risorse perché non è più AWS possibile mapparlo su un ARN valido. Il risultato è che se si elimina e si ricrea un utente o referenziato in un elemento `Principal` della policy di attendibilità, è necessario modificare il ruolo per sostituire l'ID principale non corretto con il nome ARN corretto. IAM trasforma nuovamente l'ARN nel nuovo ID principale dell'utente quando si salva la policy.

Principi fondamentali di Centro identità IAM

In Centro identità IAM, il principio di una policy basata sulle risorse deve essere definito come principale dell' Account AWS . Per specificare l'accesso, fai riferimento all'ARN del ruolo del set di autorizzazioni nel blocco delle condizioni. Per ulteriori dettagli, consulta la sezione Referenziare i set di autorizzazioni nelle policy delle risorse, in Amazon EKS e in AWS KMS nella Guida per l'utente di Centro identità IAM.

AWS STS principi di sessione utente federati

È possibile specificare le sessioni come utente federato nell'elemento `Principal` di una policy basata sulle risorse o in chiavi di condizione che supportano i principali.

> **Important**
>
> AWS consiglia di utilizzare sessioni utente AWS STS federate solo quando necessario, ad esempio quando è richiesto l'accesso da parte dell'utente root. Utilizzare invece i ruoli per delegare le autorizzazioni.

Un principale di sessione utente AWS STS federato è un principale di sessione che risulta dall'utilizzo dell' AWS STS `GetFederationToken`operazione. In questo caso, AWS STS utilizza la federazione delle identità come metodo per ottenere token di accesso temporaneo anziché utilizzare i ruoli IAM.

In AWS, gli utenti IAM o un utente Utente root dell'account AWS possono autenticarsi utilizzando chiavi di accesso a lungo termine. Per ulteriori informazioni su quali principali possono eseguire la federazione utilizzando questa operazione, consulta Confronto delle operazioni AWS STS API.

- Utente federato IAM: un utente IAM esegue la federazione utilizzando l'operazione `GetFederationToken`, che si traduce in un principale di sessione come utente federato per quell'utente IAM.
- Utente root federato: un utente root esegue la federazione usando l'operazione `GetFederationToken`, che si traduce in un principale di sessione come utente federato per quell'utente root.

Quando un utente IAM o un utente root richiede credenziali temporanee per AWS STS utilizzare questa operazione, inizia una sessione utente federata temporanea. L'ARN di questa sessione si basa sull'identità originale federata.

Per specificare l'ARN della sessione come utente federato nell'elemento `Principal`, utilizza questo formato:

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:federated-user/user-name" }
```

AWS presidi del servizio

È possibile specificare AWS i servizi nell'`Principal`elemento di una politica basata sulle risorse o in chiavi di condizione che supportano i principali. Un principale del servizio è un identificatore per un servizio.

I ruoli IAM che possono essere assunti da un AWS servizio sono chiamati ruoli di servizio. I ruoli di servizio devono includere una policy di affidabilità. Le Policy di affidabilità sono policy basate su risorse collegate a un ruolo che definisce quali principali possono assumere il ruolo. Alcuni ruoli di servizio hanno policy di affidabilità predefinite. Tuttavia, in alcuni casi, è necessario specificare il principale del servizio nella policy di affidabilità. Il service principal in una policy IAM non può esserlo`"Service": "*"`.

L'identificatore di un principale del servizio include il nome del servizio ed è solitamente nel formato seguente:

`service-name``.amazonaws.com`

Il principale del servizio è definito dal servizio. Puoi trovare il principale del servizio aprendo AWS servizi che funzionano con IAM, controllando se il servizio ha impostato Sì nella colonna Ruolo collegato ai servizi e aprendo il collegamento Sì per visualizzare la documentazione del ruolo collegato a tale servizio. Trova la sezione Autorizzazioni del ruolo collegato ai servizi per quel servizio per visualizzare il principale del servizio

L'esempio seguente mostra una policy che può essere collegata a un ruolo del servizio. Questa policy consente a due servizi, Amazon ECS e Elastic Load Balancing, di assumere il ruolo. I servizi possono eseguire qualsiasi attività concesse da una policy di autorizzazioni assegnata al ruolo (non visualizzato). Per specificare più principali del servizio, non si specificano due elementi `Service`, è possibile averne solo uno. Utilizzare invece una serie di principali del servizio come il valore di un elemento singolo `Service`.

```
"Principal": {
    "Service": [
        "ecs.amazonaws.com",
        "elasticloadbalancing.amazonaws.com"
   ]
}
```

AWS i principali del servizio nelle regioni che accettano l'adesione

Puoi lanciare risorse in diverse AWS regioni e in alcune di esse devi aderire. Per un elenco completo delle regioni a cui devi aderire, consulta Gestire AWS le regioni nella Riferimenti generali di AWSguida.

Quando un AWS servizio in una regione opt-in effettua una richiesta all'interno della stessa regione, il formato del nome principale del servizio viene identificato come la versione non regionalizzata del nome principale del servizio:

service-name.amazonaws.com

Quando un AWS servizio in una regione opt-in invia una richiesta interregionale a un'altra regione, il formato del nome principale del servizio viene identificato come la versione regionalizzata del nome principale del servizio:

service-name.*{region}*.amazonaws.com

Ad esempio, si consideri un argomento Amazon SNS situato nella Regione `ap-southeast-1` e un bucket Amazon S3 nella Regione di adesione `ap-east-1`. Supponiamo che si desideri configurare le notifiche bucket S3 per pubblicare messaggi nell'argomento SNS. Per consentire al servizio S3 di inviare messaggi all'argomento SNS, è necessario concedere l'autorizzazione `sns:Publish` del principale del servizio S3 tramite la policy di accesso basata sulle risorse dell'argomento.

Se si specifica la versione non regionalizzata del principale del servizio S3 `s3.amazonaws.com`, nella policy di accesso all'argomento, la richiesta `sns:Publish` dal bucket all'argomento avrà esito negativo. L'esempio seguente specifica il principale del servizio S3 non regionalizzato nell'elemento della policy `Principal` della policy di accesso all'argomento SNS.

```
"Principal": { "Service": "s3.amazonaws.com" }
```

Poiché il bucket si trova in una regione di adesione e la richiesta viene effettuata al di fuori della stessa regione, il principale del servizio S3 appare come nome del principale del servizio regionalizzato, `s3.ap-east-1.amazonaws.com`. È necessario utilizzare il nome principale del servizio regionalizzato quando un AWS servizio in una regione opt-in invia una richiesta a un'altra regione. Dopo aver specificato il nome del principale del servizio regionalizzato, se il bucket effettua una richiesta `sns:Publish` all'argomento SNS situato in un'altra regione, la richiesta avrà esito positivo. L'esempio seguente specifica il principale del servizio S3 regionalizzato nell'elemento della policy `Principal` della policy di accesso all'argomento SNS.

```
"Principal": { "Service": "s3.ap-east-1.amazonaws.com" }
```

Le policy di risorse o gli elenchi di autorizzazioni basati sui principali dei servizi per le richieste tra regioni da una regione di adesione a un'altra regione avranno esito positivo solo se si specifica il nome del principale del servizio regionalizzato.

> **Note**
>
> Per le policy di attendibilità dei ruoli IAM, consigliamo di utilizzare il nome del principale del servizio non regionalizzato. Le risorse IAM sono globali e quindi lo stesso ruolo può essere utilizzato in qualsiasi regione.

Tutti i principali

Puoi utilizzare un carattere jolly (*) per specificare tutti i principali nell'elemento `Principal` di una policy basata sulle risorse o nelle chiavi della condizione che supportano tali entità. Policy basate su risorse concedono le autorizzazioni e le chiavi della condizione vengono utilizzate per limitare le condizioni di un'istruzione della policy.

> **Important**
>
> Ti consigliamo di non utilizzare un carattere jolly (*) nell'elemento `Principal` di una policy basata sulle risorse con un effetto `Allow` a meno che tu non intenda concedere un accesso pubblico o anonimo. In caso contrario, specifica i principali, i servizi o gli account AWS previsti nell'elemento `Principal`, quindi limita ulteriormente l'accesso nell'elemento `Condition`. Ciò vale in special modo per le policy di attendibilità del ruolo IAM, perché consentono ad altri principali di diventare un principale nel tuo account.

Per le policy basate sulle risorse, l'utilizzo di un carattere jolly (*) con un effetto `Allow` concede l'accesso a tutti gli utenti, compresi gli utenti anonimi (accesso pubblico). Per gli utenti IAM e i principali del ruolo all'interno del tuo account non sono richieste altre autorizzazioni. Per i principali di altri account, devono inoltre disporre di autorizzazioni basate su identità nel proprio account che consentano loro di accedere alla tua risorsa. Questo è chiamato accesso tra account.

Per gli utenti anonimi, i seguenti elementi sono equivalenti:

```
"Principal": "*"
```

```
"Principal" : { "AWS" : "*" }
```

Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN.

L'esempio seguente mostra una policy basata sulle risorse che può essere utilizzata al posto di Specifica di `NotPrincipal` con `Deny` per negare esplicitamente tutti i principali, eccetto quelli specificati nell'elemento `Condition`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UsePrincipalArnInsteadOfNotPrincipalWithDeny",
      "Effect": "Deny",
      "Action": "s3:*",
      "Principal": "*",
      "Resource": [
        "arn:aws:s3:::BUCKETNAME/*",
        "arn:aws:s3:::BUCKETNAME"
      ],
      "Condition": {
        "ArnNotEquals": {
          "aws:PrincipalArn": "arn:aws:iam::444455556666:user/user-name"
        }
      }
    }
  ]
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Esempi di policy di bucket nella Guida per l'utente di Amazon Simple Storage Service (Amazon S3)
- Policy di esempio per Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service

- Policy di esempio per Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service
- Policy delle chiavi nella Guida per gli sviluppatori di AWS Key Management Service
- Identificatori di account nella Riferimenti generali di AWS
- Federazione OIDC

AWS Elementi della policy JSON: NotPrincipal

Puoi utilizzare l'`NotPrincipal`elemento per negare l'accesso a tutti i principali tranne l'utente IAM, l'utente federato, il ruolo IAM Account AWS, il AWS servizio o altro principale specificato nell'elemento. `NotPrincipal`

Puoi utilizzarlo nelle politiche basate sulle risorse per alcuni AWS servizi, inclusi gli endpoint VPC. Le policy basate su risorse sono policy che vengono incorporate direttamente in una risorsa. Non puoi utilizzare l'elemento `NotPrincipal` in una policy basata sull'identità IAM o in una policy di attendibilità del ruolo IAM.

`NotPrincipal` deve essere usato con `"Effect":"Deny"`. L'uso con `"Effect":"Allow"` non è supportato.

> **Important**
>
> Pochissimi scenari richiedono l'utilizzo di `NotPrincipal`. Si consiglia di esplorare altre opzioni di autorizzazione prima di decidere di utilizzare `NotPrincipal`. Quando utilizzi `NotPrincipal`, la risoluzione dei problemi legati agli effetti di più tipi di policy può essere difficile. Con gli operatori di condizione ARN, si consiglia invece di utilizzare la chiave di contesto `aws:PrincipalArn`. Per ulteriori informazioni, consulta Tutti i principali.

Specifica di **`NotPrincipal`** con **`Deny`**

Quando si utilizza `NotPrincipal` con `Deny`, è necessario specificare anche l'ARN dell'account del principale non rifiutato. In caso contrario, la policy potrebbe rifiutare l'accesso all'intero account contenente il principale. A seconda del servizio che si include nella policy, AWS potrebbe convalidare prima l'account e poi l'utente. Se viene valutato un utente con ruolo presunto (qualcuno che utilizza un ruolo), AWS potrebbe convalidare prima l'account, poi il ruolo e poi l'utente con il ruolo assunto. L'utente con ruolo assunto viene identificato tramite il nome della sessione del ruolo specificato quando l'utente ha assunto il ruolo. Pertanto, è fortemente consigliabile includere esplicitamente

l'ARN di un account utente oppure includere sia l'ARN di un ruolo sia l'ARN dell'account che contiene quel ruolo.

> **Important**
>
> Non utilizzare istruzioni di policy basate sulle risorse che includono un elemento di policy `NotPrincipal` con effetto `Deny` per gli utenti o i ruoli IAM ai quali è collegata una policy con limite delle autorizzazioni. L'elemento `NotPrincipal` con effetto `Deny` rifiuterà sempre qualsiasi principale IAM al quale è collegata una policy con limite delle autorizzazioni, indipendentemente dai valori specificati nell'elemento `NotPrincipal`. Ciò fa sì che alcuni utenti o ruoli IAM che altrimenti avrebbero accesso alla risorsa perdano l'accesso. Ti consigliamo di modificare le istruzioni di policy basate sulle risorse di modo che, per limitare l'accesso, utilizzino l'operatore di condizione `ArnNotEquals` con la chiave di contesto `aws:PrincipalArn` anziché l'elemento `NotPrincipal`. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta la pagina Limiti delle autorizzazioni per le entità IAM.

> **Note**
>
> Come best practice, si dovrebbero includere gli ARN per l'account nella policy. Alcuni servizi richiedono l'ARN dell'account, anche se questo non è obbligatorio in tutti i casi. Le policy esistenti senza l'ARN richiesto continueranno a funzionare, ma le nuove policy che includono tali servizi devono soddisfare questo requisito. IAM non tiene traccia di questi servizi e pertanto consiglia di includere sempre l'ARN dell'account.

I seguenti esempi mostrano come utilizzare `NotPrincipal` e `"Effect": "Deny"` nella stessa istruzione della policy in modo efficiente.

Example Esempio di utente IAM nello stesso account o in un account differente

Nell'esempio seguente, a tutti i responsabili tranne l'utente denominato Bob in 444455556666 viene esplicitamente negato l'accesso a Account AWS una risorsa. Nota che, come best practice, l'`NotPrincipal`elemento contiene l'ARN sia dell'utente Bob Account AWS che di quello a cui Bob appartiene ()`arn:aws:iam::444455556666:root`. Se l'`NotPrincipal`elemento conteneva solo l'ARN di Bob, l'effetto della policy potrebbe essere quello di negare esplicitamente l'accesso all'ARN Account AWS che contiene l'utente Bob. In alcuni casi, un utente non può avere più autorizzazioni

rispetto al rispettivo account padre, quindi se all'account di Bob viene esplicitamente rifiutato l'accesso, Bob potrebbe non essere in grado di accedere alla risorsa.

Questo esempio funziona come previsto quando fa parte di una dichiarazione politica in una politica basata sulle risorse allegata a una risorsa uguale o diversa (non 444455556666). Account AWS Questo esempio di per sé non concede l'accesso a Bob, omette solo Bob dall'elenco di principali esplicitamente rifiutati. Per consentire a Bob di accedere alla risorsa, un'altra istruzione della policy deve permettere esplicitamente l'accesso tramite `"Effect": "Allow"`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Deny",
        "NotPrincipal": {"AWS": [
            "arn:aws:iam::444455556666:user/Bob",
            "arn:aws:iam::444455556666:root"
        ]},
        "Action": "s3:*",
        "Resource": [
            "arn:aws:s3:::BUCKETNAME",
            "arn:aws:s3:::BUCKETNAME/*"
        ]
    }]
}
```

Example Esempio di ruolo IAM nello stesso account o in un account differente

Nell'esempio seguente, a tutti i principali, tranne l'utente con ruolo assunto denominato in 444455556666, viene esplicitamente negato l'accesso cross-account-audit-app a Account AWS una risorsa. Come procedura ottimale, l'`NotPrincipal`elemento contiene l'ARN dell'utente assunto (cross-account-audit-app), del ruolo (-role) e del ruolo a cui cross-account-read-only appartiene il ruolo (Account AWS 444455556666). Se nell'elemento `NotPrincipal` mancasse l'ARN del ruolo, l'effetto della policy potrebbe essere di rifiutare esplicitamente l'accesso al ruolo. Analogamente, se nell'elemento `NotPrincipal` mancasse l'ARN dell' Account AWS a cui appartiene il ruolo, l'effetto della policy potrebbe essere di rifiutare esplicitamente l'accesso all' Account AWS e a tutte le entità in tale account. In alcuni casi, gli utenti con ruolo presunto non possono avere più autorizzazioni del ruolo principale e i ruoli non possono avere più autorizzazioni del ruolo principale Account AWS, quindi quando al ruolo o all'account viene negato esplicitamente l'accesso, l'utente assunto potrebbe non essere in grado di accedere alla risorsa.

Questo esempio funziona come previsto quando fa parte di un'informativa in un criterio basato sulle risorse collegato a una risorsa in un altro (non 444455556666). Account AWS Questo esempio di per sé non consente l'accesso all'utente che assume il ruolo cross-account-audit-app, ma si limita a omettere cross-account-audit-app dall'elenco dei principali che vengono esplicitamente negati. Per cross-account-audit-app consentire l'accesso alla risorsa, un'altra dichiarazione politica deve consentire esplicitamente l'utilizzo di. `"Effect": "Allow"`

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Deny",
        "NotPrincipal": {"AWS": [
            "arn:aws:sts::444455556666:assumed-role/cross-account-read-only-role/cross-
account-audit-app",
            "arn:aws:iam::444455556666:role/cross-account-read-only-role",
            "arn:aws:iam::444455556666:root"
        ]},
        "Action": "s3:*",
        "Resource": [
            "arn:aws:s3:::Bucket_AccountAudit",
            "arn:aws:s3:::Bucket_AccountAudit/*"
        ]
    }]
}
```

Quando si specifica una sessione con assunzione di ruolo in un elemento `NotPrincipal`, non è possibile utilizzare un carattere jolly (*) per indicare "tutte le sessioni". Le entità devono sempre fare riferimento a una sessione specifica.

Elementi delle policy JSON IAM: Action

L'elemento `Action` descrive l'operazione o le operazioni specifiche che saranno concesse o negate. Le istruzioni devono includere un elemento `Action` o un elemento `NotAction`. Ogni AWS servizio dispone di un proprio set di azioni che descrivono le attività che è possibile eseguire con tale servizio. Ad esempio, l'elenco delle azioni per Amazon S3 è disponibile in Specificying Permissions in a Policy nellaAmazon Simple Storage Service User Guide, l'elenco delle azioni per Amazon EC2 è disponibile nel riferimento alle API di Amazon EC2 e l'elenco delle azioni per AWS Identity and Access Management è disponibile nel riferimento alle API IAM. Per trovare l'elenco delle operazioni per altri servizi, consulta la documentazione di riferimento alle API per il servizio.

È possibile specificare un valore utilizzando uno spazio dei nomi come prefisso dell'operazione (`iam`, `ec2` `sqs`, `sns`, `s3`, ecc.) seguito dal nome dell'operazione da consentire o negare. Il nome deve corrispondere a un'operazione che è supportata dal servizio. Il prefisso e il nome dell'operazione non fanno distinzione tra maiuscole e minuscole. Ad esempio, `iam:ListAccessKeys` è equivalente a `IAM:listaccesskeys`. I seguenti esempi mostrano gli elementi `Action` per diversi servizi.

Operazione di Amazon SQS

```
"Action": "sqs:SendMessage"
```

Operazione Amazon EC2

```
"Action": "ec2:StartInstances"
```

Operazione IAM

```
"Action": "iam:ChangePassword"
```

Operazioni di Amazon S3

```
"Action": "s3:GetObject"
```

Puoi specificare valori multipli per l'elemento `Action`.

```
"Action": [ "sqs:SendMessage", "sqs:ReceiveMessage", "ec2:StartInstances",
 "iam:ChangePassword", "s3:GetObject" ]
```

Puoi utilizzare un carattere jolly (*) per consentire l'accesso a tutte le azioni offerte dallo specifico prodotto. AWS Ad esempio, il seguente elemento `Action` si applica a tutte le operazioni S3.

```
"Action": "s3:*"
```

Puoi anche utilizzare un carattere jolly (*) come parte del nome dell'operazione. Ad esempio, il seguente elemento `Action` si applica a tutte le operazioni IAM che includono la stringa `AccessKey`, incluso `CreateAccessKey`, `DeleteAccessKey`, `ListAccessKeys` e `UpdateAccessKey`.

```
"Action": "iam:*AccessKey*"
```

Alcuni servizi ti consentono di limitare le operazioni disponibili. Ad esempio, Amazon SQS consente di rendere disponibile solo un sottoinsieme di tutte le operazioni Amazon SQS possibili. In questo caso, il carattere jolly * non ti permette il controllo completo della coda; ti permette solo il sottoinsieme di operazioni che hai condiviso. Per ulteriori informazioni, consulta Informazioni sulle autorizzazioni nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Elementi delle policy JSON IAM: NotAction

`NotAction` è un elemento di policy avanzato che corrisponde esplicitamente a tutte le operazioni tranne quelle specificamente elencate. L'utilizzo di `NotAction` può determinare una policy più breve dal momento che è possibile elencare solo poche operazioni che non devono corrispondere, anziché includere un lungo elenco di operazioni che devono corrispondere. Le azioni specificate in non `NotAction` sono influenzate dall'`Deny`effetto `Allow` o contenuto in una dichiarazione politica. Questo significa a sua volta che tutte le operazioni o i servizi applicabili che non sono elencati sono consentiti se utilizzi l'effetto `Allow`. Inoltre, tali operazioni o servizi non elencati vengono negati se utilizzi l'effetto `Deny`. Quando utilizzi `NotAction` con l'elemento `Resource`, fornisci l'ambito della policy. In questo modo si AWS determinano le azioni o i servizi applicabili. Per ulteriori informazioni, consulta la policy di esempio seguente.

NotAction con Allow

È possibile utilizzare l'`NotAction`elemento in un'istruzione con `"Effect": "Allow"` per fornire l'accesso a tutte le azioni di un AWS servizio, ad eccezione delle azioni specificate in`NotAction`. È possibile utilizzarlo con l'elemento `Resource` per fornire l'ambito della policy, limitando le operazioni consentite a quelle che possono essere eseguite sulla risorsa specificata.

L'esempio seguente consente agli utenti di accedere a tutte le operazioni Amazon S3 che possono essere eseguite su qualsiasi risorsa S3 eccetto l'eliminazione di un bucket. Questo esempio non consente agli utenti di utilizzare l'operazione dell'API S3 `ListAllMyBuckets`, perché tale operazione richiede la risorsa "*". Inoltre, questa policy non consente operazioni in altri servizi, perché le operazioni di altri servizi non sono applicabili alle risorse S3.

```
"Effect": "Allow",
"NotAction": "s3:DeleteBucket",
"Resource": "arn:aws:s3:::*",
```

È possibile che talvolta si desideri consentire l'accesso a un numero elevato di operazioni. Utilizzando l'elemento `NotAction` si inverte efficacemente l'istruzione, determinando un elenco più breve di

operazioni. Ad esempio, poiché AWS dispone di così tanti servizi, potresti voler creare una policy che consenta all'utente di fare tutto tranne accedere alle azioni IAM.

L'esempio seguente consente agli utenti di accedere a tutte le azioni in tutti i AWS servizi tranne IAM.

```
"Effect": "Allow",
"NotAction": "iam:*",
"Resource": "*"
```

Va prestata attenzione all'utilizzo dell'elemento `NotAction` e `"Effect": "Allow"` nella stessa istruzione o in un'istruzione diversa nella policy. `NotAction` corrisponde a tutti i servizi e le operazioni che non sono esplicitamente elencati o applicabili alla risorsa specificata e può finire col concedere agli utenti più autorizzazioni del previsto.

NotAction con Deny

Puoi utilizzare l'elemento `NotAction` in un'istruzione con `"Effect": "Deny"` per negare l'accesso a tutte le risorse elencate tranne le operazioni specificate nell'elemento `NotAction`. Questa combinazione non consente gli elementi elencati ma invece nega esplicitamente le operazioni non elencate. Devi comunque consentire le operazioni che desideri consentire.

Il seguente esempio condizionale nega l'accesso alle operazioni non IAM se l'utente non ha eseguito l'accesso utilizzando l'autenticazione MFA. Se l'utente ha eseguito l'accesso con l'autenticazione MFA, il test `"Condition"` non riesce e l'istruzione `"Deny"` finale non produce effetti. Nota, tuttavia, che questa istruzione non concederebbe all'utente l'accesso ad alcuna operazione, ma negherebbe solamente in modo esplicito tutte le altre operazioni eccetto le operazioni IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "DenyAllUsersNotUsingMFA",
        "Effect": "Deny",
        "NotAction": "iam:*",
        "Resource": "*",
        "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}}
    }]
}
```

Per una policy di esempio che nega l'accesso alle operazioni al di fuori di regioni specifiche, ad eccezione delle operazioni di servizi specifici, consulta AWS: nega l'accesso in AWS base alla regione richiesta.

Elementi delle policy JSON IAM: Resource

L'elemento `Resource` specifica l'oggetto o gli oggetti coperti dall'istruzione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. È possibile specificare una risorsa utilizzando un ARN. Per ulteriori informazioni sul formato di ARN, consulta ARN IAM.

Ogni servizio possiede il proprio gruppo di risorse. Anche se utilizzi sempre un ARN per specificare una risorsa, i dettagli dell'ARN per una risorsa dipendono dal servizio e dalla risorsa. Per informazioni su come specificare una risorsa, consulta la documentazione relativa al servizio per la quale desideri scrivere un'istruzione.

> **Note**
>
> Alcuni servizi non consentono di specificare le operazioni per le singole risorse; invece, ogni operazione elencata nell'elemento `Action` o `NotAction` si applica a tutte le risorse in quel servizio. In questi casi, è possibile utilizzare il carattere jolly `*` nell'elemento `Resource`.

L'esempio seguente si riferisce a una determinata coda Amazon SQS.

```
"Resource": "arn:aws:sqs:us-east-2:account-ID-without-hyphens:queue1"
```

L'esempio seguente si riferisce all'utente IAM denominato Bob in un Account AWS.

> **Note**
>
> Nell'elemento `Resource`, il nome utente IAM prevede una distinzione tra lettere minuscole e maiuscole.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/Bob"
```

Utilizzo di caratteri jolly negli ARN delle risorse

È possibile utilizzare caratteri jolly come parte dell'ARN della risorsa. Puoi usare caratteri jolly (* e ?) all'interno dei segmenti dell'ARN (le parti separate da due punti) per rappresentare qualsiasi combinazione di caratteri con un asterisco (*) e qualsiasi carattere singolo con un punto interrogativo (?). È possibile utilizzare più caratteri * o ? in ogni segmento. Se il carattere jolly asterisco (*) è l'ultimo carattere del segmento dell'ARN di una risorsa, può espandersi fino a superare i limiti dei due

punti. Consigliamo di utilizzare i caratteri jolly (* e ?) all'interno dei segmenti dell'ARN separati da due punti.

> **Note**
>
> Non puoi utilizzare un carattere jolly nel segmento di servizio che identifica il AWS prodotto. Per ulteriori informazioni sui segmenti degli ARN, consulta Amazon Resource Names (ARN)

L'esempio seguente si riferisce a tutti gli utenti IAM il cui percorso è `/accounting`.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/accounting/*"
```

L'esempio seguente si riferisce a tutti gli elementi all'interno di un determinato bucket Amazon S3.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

Il carattere asterisco (*) può espandersi per sostituire tutto all'interno di un segmento, inclusi caratteri come una barra (/) che potrebbero sembrare un delimitatore all'interno di un determinato spazio dei nomi del servizio. Ad esempio, considera il seguente ARN di Amazon S3 come la stessa logica di espansione con caratteri jolly si applica a tutti i servizi.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/test/*"
```

I caratteri jolly nell'ARN si applicano a tutti i seguenti oggetti nel bucket, non solo al primo oggetto elencato.

```
DOC-EXAMPLE-BUCKET/1/test/object.jpg
DOC-EXAMPLE-BUCKET/1/2/test/object.jpg
DOC-EXAMPLE-BUCKET/1/2/test/3/object.jpg
DOC-EXAMPLE-BUCKET/1/2/3/test/4/object.jpg
DOC-EXAMPLE-BUCKET/1///test///object.jpg
DOC-EXAMPLE-BUCKET/1/test/.jpg
DOC-EXAMPLE-BUCKET//test/object.jpg
DOC-EXAMPLE-BUCKET/1/test/
```

Considera gli ultimi due oggetti dell'elenco precedente. Il nome di un oggetto Amazon S3 può iniziare o terminare validamente con il carattere barra (/) del delimitatore convenzionale. Mentre "/" funziona come delimitatore, non vi è alcun significato specifico quando questo carattere viene

utilizzato all'interno di una risorsa ARN. Viene trattato come qualsiasi altro carattere valido. L'ARN non corrisponde ai seguenti oggetti:

```
DOC-EXAMPLE-BUCKET/1-test/object.jpg
DOC-EXAMPLE-BUCKET/test/object.jpg
DOC-EXAMPLE-BUCKET/1/2/test.jpg
```

Specifica di più risorse

Puoi specificare più risorse. L'esempio seguente si riferisce a due tabelle DynamoDB.

```
"Resource": [
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/books_table",
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/magazines_table"
]
```

Utilizzo delle variabili delle policy negli ARN delle risorse

Nell'elemento `Resource`, puoi utilizzare le variabili di policy JSON nella parte dell'ARN che identifica la risorsa specifica (ovvero nella parte finale di ARN). Ad esempio, puoi utilizzare la chiave `{aws:username}` come parte di una risorsa ARN per indicare che l'attuale nome dell'utente deve essere incluso come parte del nome della risorsa. L'esempio seguente mostra come puoi utilizzare la chiave `{aws:username}` in un elemento `Resource`. La policy consente l'accesso a una tabella Amazon DynamoDB che corrisponde al nome dell'utente corrente.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "dynamodb:*",
    "Resource": "arn:aws:dynamodb:us-east-2:account-id:table/${aws:username}"
  }
}
```

Per ulteriori informazioni sulle variabili di policy JSON, consultare la pagina Elementi delle policy IAM: variabili e tag.

Elementi delle policy JSON IAM: NotResource

`NotResource` è un elemento della policy avanzato che corrisponde esplicitamente a tutte le risorse tranne quelle specificate. L'utilizzo di `NotResource` può risultare in una policy di durata inferiore

elencando solo poche risorse che non devono corrispondere, anziché includere un lungo elenco di risorse che corrisponderanno. Ciò è particolarmente utile per le policy che si applicano all'interno di un singolo servizio AWS .

Ad esempio, immaginate di disporre di un gruppo denominato `HRPayroll`. I membri di `HRPayroll` non devono avere il permesso di accedere a qualsiasi risorsa Amazon S3 ad eccezione della cartella `Payroll` nel bucket `HRBucket`. La policy seguente rifiuta esplicitamente l'accesso a tutte le risorse Amazon S3 eccetto a quelle elencate. Tuttavia, questa policy non concede all'utente l'accesso a nessuna risorsa.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "s3:*",
    "NotResource": [
      "arn:aws:s3:::HRBucket/Payroll",
      "arn:aws:s3:::HRBucket/Payroll/*"
    ]
  }
}
```

Di solito, per negare esplicitamente l'accesso a una risorsa è necessario scrivere una policy che utilizza `"Effect":"Deny"` e che include un elemento `Resource` che elenca ogni cartella individualmente. Tuttavia, in tal caso, ogni volta che aggiungi una cartella a `HRBucket` o una risorsa a Amazon S3 che non deve essere accessibile, è necessario aggiungere il suo nome all'elenco in `Resource`. Se si utilizza invece un elemento `NotResource`, agli utenti viene automaticamente negato l'accesso a nuove cartelle a meno che non si aggiungano i nomi delle cartelle all'elemento `NotResource`.

Quando si utilizza `NotResource`, è necessario tenere presente che le risorse specificate in questo elemento sono le uniche risorse a non essere limitate. Questo, a sua volta, limita tutte le risorse che si applicano all'operazione. Nell'esempio precedente, la policy riguarda solo le operazioni di Amazon S3 e quindi solo le risorse di Amazon S3. Se l'operazione avesse incluso anche operazioni di Amazon EC2, la policy non avrebbe rifiutato l'accesso a risorse EC2. Per sapere quali azioni in un servizio consentono di specificare l'ARN di una risorsa, consulta Azioni, risorse e chiavi AWS di condizione per i servizi.

NotResource con altri elementi

Non bisognerebbe mai utilizzare insieme gli elementi `"Effect": "Allow"`, `"Action": "*"` e `"NotResource": "arn:aws:s3:::HRBucket"`. Questa affermazione è molto pericolosa, perché consente tutte le azioni AWS su tutte le risorse tranne il bucket `HRBucket` S3. Ciò consentirebbe addirittura a un utente di aggiungere al proprio profilo una policy che gli consenta di accedere a `HRBucket`. Non bisogna farlo.

Va prestata attenzione all'utilizzo dell'elemento `NotResource` e `"Effect": "Allow"` nella stessa istruzione o in un'istruzione diversa nella policy. `NotResource` consente tutti i servizi e risorse che non sono elencati in modo esplicito e può concedere agli utenti più autorizzazioni del previsto. L'utilizzo dell'elemento `NotResource` e `"Effect": "Deny"` nella stessa istruzione nega i servizi e le risorse che non sono elencati in modo esplicito.

Elementi delle policy JSON IAM: Condition

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni di attivazione di una policy. L'elemento `Condition` è facoltativo. Nell'elemento `Condition` è possibile creare espressioni in cui utilizzare operatori condizionali (uguale a, meno di, e altri) per confrontare le chiavi di contesto e i valori della policy rispetto alle chiavi e ai valori del contesto della richiesta. Per ulteriori informazioni sul contesto della richiesta, consultare Richiesta.

```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

La chiave di contesto specificata in una condizione della policy può essere una chiave di contesto della condizione globale o una chiave di condizione specifica del servizio. Le chiavi di contesto della condizione globale presentano il prefisso `aws:`. Le chiavi di contesto specifiche del servizio presentano il prefisso del servizio. Ad esempio, Amazon EC2 consente di scrivere una condizione utilizzando la chiave di contesto `ec2:InstanceType`, univoca per quel servizio. Per visualizzare le chiavi di contesto IAM specifiche del servizio con il prefisso `iam:`, consulta chiavi contestuali IAM e AWS STS condition.

I nomi delle chiavi di contesto non fanno distinzione tra maiuscole e minuscole. Ad esempio, se si include la chiave di contesto `aws:SourceIP` è identico al test per la chiave `AWS:SourceIp`. La distinzione tra maiuscole e minuscole dei valori delle chiavi di contesto dipende dall'operatore di condizione utilizzato. Ad esempio, la seguente condizione include l'operatore `StringEquals` per rendere possibile la corrispondenza solo delle richieste effettuate da `johndoe`. Agli utenti denominati `JohnDoe` viene negato l'accesso.

```
"Condition" : { "StringEquals" : { "aws:username" : "johndoe" }}
```

Le seguenti condizione utilizza l'operatore `StringEqualsIgnoreCase` per corrispondere agli utenti denominati `johndoe` o `JohnDoe`.

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "johndoe" }}
```

Alcune chiavi di contesto supportano le coppie chiave-valore che consentono di specificare parte del nome della chiave. Gli esempi includono la chiave di `aws:RequestTag/tag-key`contesto AWS KMS `kms:EncryptionContext:encryption_context_key`, la e la chiave di `ResourceTag/tag-key`contesto supportate da più servizi.

- Se utilizzi la chiave di contesto `ResourceTag/tag-key` per un servizio come Amazon EC2, devi specificare un nome di chiave per la `tag-key`.
- I nomi delle chiavi non fanno distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe.
- AWS i servizi che supportano questi attributi potrebbero consentire di creare più nomi di chiavi che differiscono solo in base alle maiuscole e alle minuscole. Ad esempio, è possibile applicare un tag a un'istanza Amazon EC2 con `ec2=test1` e `EC2=test2`. Quando utilizzi una condizione come `"aws:ResourceTag/EC2": "test1"` per consentire l'accesso alla risorsa, il nome della chiave corrisponde a entrambi i tag, ma solo a un valore. Questo può causare errori di condizione imprevisti.

> **Important**
>
> Come best practice, verifica che i membri del tuo account seguano una convenzione di denominazione coerente quando assegnano nomi agli attributi di coppie chiave-valore. Alcuni esempi includono tag o contesti di crittografia AWS KMS . È possibile imporlo utilizzando la chiave di `aws:TagKeys`contesto per l'etichettatura o la `kms:EncryptionContextKeys`per il contesto di AWS KMS crittografia.

- Per un elenco di tutti gli operatori di condizione e per una descrizione del funzionamento di ciascun operatore, consulta Operatori di condizione

- Se non è diversamente specificato, tutte le chiavi di contesto possono avere valori multipli. Per ulteriori informazioni sulla gestione delle chiavi di contesto che dispongono di più valori, consulta Chiavi di contesto multivalore
- Per un elenco di tutte le chiavi di contesto disponibili a livello globale, consulta AWS chiavi di contesto della condizione globale.
- Per le chiavi di contesto delle condizioni definite da ciascun servizio, consulta Azioni, risorse e chiavi di condizione per i AWS servizi.

Il contesto della richiesta

Quando un preside effettua una richiesta a AWS, AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta. Le informazioni vengono utilizzate per valutare e autorizzare la richiesta. È possibile utilizzare l'elemento `Condition` di una policy JSON per testare chiavi di contesto specifiche rispetto al contesto della richiesta. Ad esempio, puoi creare una policy che utilizzi la chiave aws: CurrentTime context per consentire a un utente di eseguire azioni solo entro un intervallo di date specifico.

Quando viene inviata una richiesta, AWS valuta ogni chiave di contesto nella policy e restituisce il valore true, false, not present e occasionalmente null (una stringa di dati vuota). Una chiave di contesto che non è presente nella richiesta è considerata una mancata corrispondenza. Ad esempio, la policy seguente ti consente di rimuovere il tuo dispositivo di autenticazione a più fattori (MFA), ma solo se è stato effettuato l'accesso utilizzando MFA nell'ultima ora (3.600 secondi).

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowRemoveMfaOnlyIfRecentMfa",
        "Effect": "Allow",
        "Action": [
            "iam:DeactivateMFADevice"
        ],
        "Resource": "arn:aws:iam::*:user/${aws:username}",
        "Condition": {
            "NumericLessThanEquals": {"aws:MultiFactorAuthAge": "3600"}
        }
    }
}
```

Il contesto della richiesta può restituire i seguenti valori:

- True: se il richiedente ha effettuato l'accesso utilizzando MFA nell'ultima ora o meno, la condizione restituisce true.
- False: se il richiedente ha effettuato l'accesso utilizzando MFA più di un'ora fa, la condizione restituisce false.
- Non presente: se il richiedente ha effettuato una richiesta utilizzando le proprie chiavi di accesso utente IAM nell' AWS API AWS CLI or, la chiave non è presente. In questo caso, la chiave non è presente e non viene restituita la corrispondenza.
- Null: per le chiavi di contesto definite dall'utente, ad esempio passando tag in una richiesta, è possibile includere una stringa vuota. In questo caso, il valore nel contesto della richiesta è null. Un valore nullo potrebbe restituire true in alcuni casi. Ad esempio, se utilizzi l'operatore di condizione multivalore `ForAllValues` con la chiave di contesto `aws:TagKeys`, è possibile ottenere risultati imprevisti se il contesto della richiesta restituisce null. Per ulteriori informazioni, consulta aws: TagKeys andChiavi di contesto multivalore.

Il blocco condizione

L'esempio seguente mostra il formato di base di un elemento `Condition`:

```
"Condition": {"StringLike": {"s3:prefix": ["janedoe/*"]}}
```

Un valore dalla richiesta è rappresentato da una chiave di contesto, in questo caso `s3:prefix`. Il valore della chiave di contesto viene confrontato con un valore specificato come valore letterale, ad esempio `janedoe/*`. Il tipo di confronto da eseguire viene specificato dall'operatore di condizione (in questo caso, `StringLike`). Puoi creare condizioni che confrontano stringhe, date, numeri e altro ancora, utilizzando tipiche comparazioni booleane come ad esempio "uguale a", "maggiore di" e "minore di". Se utilizzi operatori stringa o operatori ARN, puoi utilizzare una variabile di policy nel valore della chiave di contesto. L'esempio seguente include la variabile `aws:username`.

```
"Condition": {"StringLike": {"s3:prefix": ["${aws:username}/*"]}}
```

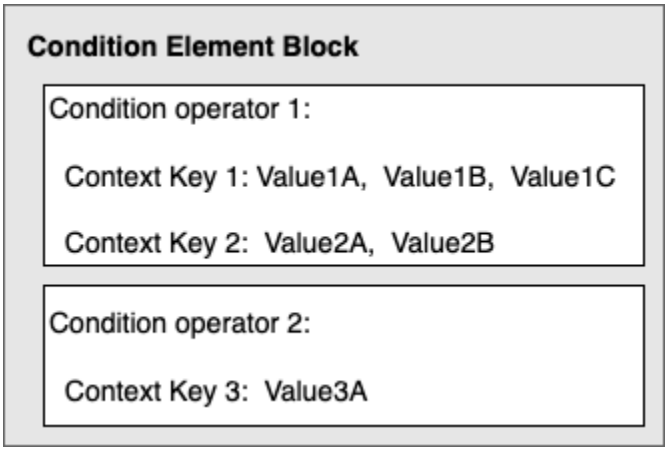
In alcuni casi, le chiavi di contesto possono contenere più valori. Ad esempio, una richiesta ad Amazon DynamoDB potrebbe richiedere la restituzione o l'aggiornamento di più attributi di una tabella. Una policy per l'accesso alle tabelle di DynamoDB può includere la chiave `dynamodb:Attributes` che contiene tutti gli attributi elencati nella richiesta. Puoi testare i vari attributi nella richiesta a fronte di un elenco di attributi consentiti in una policy, utilizzando

operatori predefiniti nell'elemento `Condition`. Per ulteriori informazioni, consulta Chiavi di contesto multivalore.

Quando la politica viene valutata durante una richiesta, AWS sostituisce la chiave con il valore corrispondente della richiesta. (In questo esempio, AWS utilizzerebbe la data e l'ora della richiesta.) Dopo la valutazione della condizione, viene restituito un risultato True o False, che viene poi utilizzato per decidere se la policy nel suo complesso deve consentire o rifiutare la richiesta.

Valori multipli in una condizione

Un elemento `Condition` può contenere più operatori di condizioni, ciascuno delle quali può includere a sua volta più coppie chiave-valore. L'immagine seguente illustra questo scenario.



Per ulteriori informazioni, consulta Chiavi di contesto multivalore.

Elementi della policy JSON IAM: operatori di condizione

Utilizzare gli operatori di condizione nell'elemento `Condition` per confrontare chiave e valore nella policy con i valori nel contesto della richiesta. Per ulteriori informazioni sull'elemento `Condition`, consultare Elementi delle policy JSON IAM: Condition.

L'operatore di condizione che è possibile utilizzare in una policy dipende dalla chiave di condizione scelta. È possibile scegliere una chiave di condizione globale o una chiave di condizione specifica del servizio. Per informazioni su quale operatore di condizione è possibile utilizzare per una chiave di condizione globale, consultare AWS chiavi di contesto della condizione globale. Per sapere quale operatore di condizione è possibile utilizzare per una chiave di condizione specifica del servizio, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e scegli il servizio che desideri visualizzare.

> **Important**
>
> Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono e la condizione è false. Se la condizione di policy richiede che la chiave sia non abbinata, ad esempio `StringNotLike` o `ArnNotLike` e la chiave giusta non è presente, la condizione è true. Questa logica si applica a tutti gli operatori di condizione tranne... IfExistse Null check. Questi operatori testano se la chiave è presente (esiste) nel contesto della richiesta.

Gli operatori di condizione possono essere raggruppati nelle seguenti categorie:

- Stringa
- Numerici
- Data e ora
- Booleano
- Binary
- IP address (Indirizzo IP)
- Amazon Resource Name (ARN) (disponibile solo per alcuni servizi)
- ... IfExists(verifica se il valore della chiave esiste come parte di un altro controllo)
- Verifica Null (controlla se il valore della chiave esiste come controllo autonomo)

Operatori di condizione stringa

Gli operatori di condizioni stringa consentono di creare elementi `Condition` che limitano l'accesso in base al confronto con una chiave con un valore di stringa.

Operatore di condizione	Descrizione
`StringEquals`	Corrispondenza esatta, con distinzione maiuscole/minuscole
`StringNotEquals`	Corrispondenza negativa
`StringEqualsIgnoreCase`	Corrispondenza esatta, senza distinzione maiuscole/minuscole

Operatore di condizione	Descrizione
`StringNotEqualsIgnoreCase`	Corrispondenza negativa, senza distinzione maiuscole/minuscole
`StringLike`	Corrispondenza con distinzione maiuscole/minuscole. I valori possono includere una corrispondenza con più caratteri jolly (*) e un singolo carattere jolly (?) in qualsiasi punto della stringa. Per ottenere corrispondenze di stringhe parziali devi specificare caratteri jolly. **Note** Se una chiave contiene più valori, `StringLike` può essere qualificato con gli operatori su set: `ForAllValues:StringLike` e `ForAnyValue:StringLike`. Per ulteriori informazioni, consulta Chiavi di contesto multivalore.
`StringNotLike`	Corrispondenza negativa, con distinzione maiuscole/minuscole. I valori possono includere una corrispondenza con più caratteri jolly (*) o un singolo carattere jolly (?) in qualsiasi punto della stringa.

Ad esempio, l'istruzione seguente contiene un elemento `Condition` che utilizza la chiave `aws:PrincipalTag` per specificare che il principale che effettua la richiesta deve essere contrassegnato con la categoria di processo `iamuser-admin`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::account-id:user/*",
    "Condition": {"StringEquals": {"aws:PrincipalTag/job-category": "iamuser-admin"}}
  }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. In questo esempio, la chiave `aws:PrincipalTag/job-category` è presente nel contesto della richiesta se il principale utilizza un utente IAM con tag collegati. È inclusa anche per un principale che utilizza un ruolo IAM con tag collegati o tag di sessione. Se un utente senza il tag tenta di visualizzare o modificare una chiave di accesso, la condizione restituisce `false` e la richiesta viene negata implicitamente da questa istruzione.

Puoi utilizzare una variabile di policy con l'operatore di condizione `String`.

Nell'esempio seguente viene utilizzato l'operatore di condizione `StringLike` per eseguire il confronto con una variabile di policy per creare una policy che consente a un utente IAM di utilizzare la console Amazon S3 per gestire la propria "directory principale" in un bucket Amazon S3. La policy consente le operazioni specificate in un bucket S3 a condizione che `s3:prefix` corrisponda a uno qualsiasi dei modelli specificati.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::BUCKET-NAME",
      "Condition": {"StringLike": {"s3:prefix": [
        "",
        "home/",
        "home/${aws:username}/"
      ]}}
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}",
        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}/*"
```

```
      ]
    }
  ]
}
```

Per un esempio di politica che mostra come utilizzare l'`Condition`elemento per limitare l'accesso alle risorse in base a un ID applicazione e un ID utente per la federazione OIDC, vedere. Amazon S3: consente agli utenti di Amazon Cognito di accedere a oggetti nel relativo bucket

Corrispondenza dei caratteri jolly

Gli operatori di condizioni di stringa eseguono una corrispondenza senza modello che non impone un formato predefinito. Gli operatori di condizione ARN e Date sono un sottoinsieme di operatori di stringa che impongono una struttura al valore della chiave della condizione. Quando si utilizzano StringNotLike gli operatori StringLike or per le corrispondenze parziali di stringhe di un ARN o di una data, la corrispondenza ignora quale parte della struttura è contrassegnata da caratteri jolly.

Ad esempio, le seguenti condizioni cercano una corrispondenza parziale di un ARN utilizzando operatori di condizioni diversi.

Quando ArnLike viene utilizzata, le parti partition, service, account-id, resource-type e resource-id parziale dell'ARN devono corrispondere esattamente all'ARN nel contesto della richiesta. La corrispondenza parziale è consentita solo per la regione e il percorso della risorsa.

```
"Condition": {"ArnLike": {"aws:SourceArn": "arn:aws:cloudtrail:*:111122223333:trail/
*"}}
```

Quando StringLike viene utilizzato al posto di ArnLike, matching ignora la struttura ARN e consente la corrispondenza parziale, indipendentemente dalla parte che è stata selezionata come wildcard.

```
"Condition": {"StringLike": {"aws:SourceArn": "arn:aws:cloudtrail:*:111122223333:trail/
*"}}
```

ARN	ArnLike	StringLike
arn:aws:cloudtrail:us-west-2:111122223333:trail/finance	Match	Match

ARN	ArnLike	StringLike
arn:aws:cloudtrail:us-east-2:111122223333:trail/finance/archive	Match	Match
arn:aws:cloudtrail:us-east-2:444455556666:user/111122223333:trail/finance	Nessuna corrispondenza	Match

Operatori di condizione numerici

Gli operatori di condizione numerici consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un valore intero o decimale.

Operatore di condizione	Descrizione
`NumericEquals`	Corrispondenza
`NumericNotEquals`	Corrispondenza negativa
`NumericLessThan`	Corrispondenza "Minore di"
`NumericLessThanEquals`	Corrispondenza "Minore di o uguale a"
`NumericGreaterThan`	Corrispondenza "Maggiore di"
`NumericGreaterThanEquals`	Corrispondenza "Maggiore di o uguale a"

Ad esempio, la seguente istruzione contiene un elemento `Condition` che utilizza l'operatore di condizione `NumericLessThanEquals` con la chiave `s3:max-keys` per specificare che il richiedente può elencare fino a oggetti in `example_bucket` alla volta.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket",
    "Condition": {"NumericLessThanEquals": {"s3:max-keys": "10"}}
  }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. In questo esempio, la chiave `s3:max-keys` è sempre presente nella richiesta quando si esegue l'operazione `ListBucket`. Se questa policy consentiva tutte le operazioni Amazon S3, saranno consentite solo le operazioni che includono la chiave di contesto `max-keys` con un valore minore o uguale a 10.

Puoi utilizzare una variabile di policy con l'operatore di condizione `Numeric`.

Operatori di condizione data

Gli operatori di condizione data consentono di creare elementi `Condition` che limitano l'accesso in base al confronto con una chiave con un valore data/ora. Utilizza questi operatori di condizione con la chiave `aws:CurrentTime` o la chiave `aws:EpochTime`. È necessario specificare i valori di data/ora con una delle implementazioni W3C dei formati di data ISO 8601 o con tempo epoca (UNIX epoch).

> **Note**
>
> Per gli operatori di condizione data, non sono ammessi caratteri jolly.

Operatore di condizione	Descrizione
`DateEquals`	Corrispondenza con una data specifica
`DateNotEquals`	Corrispondenza negativa
`DateLessThan`	Corrispondenza prima di una determinata data e ora
`DateLessThanEquals`	Corrispondenza a una determinata data e ora
`DateGreaterThan`	Corrispondenza dopo una determinata data e ora

Operatore di condizione	Descrizione
`DateGreaterThanEqu als`	Corrispondenza a una determinata data e ora o successiva

Ad esempio, l'istruzione seguente contiene un elemento `Condition` che utilizza l'operatore di condizione `DateGreaterThan` con la chiave `aws:TokenIssueTime`. Questa condizione specifica che le credenziali di sicurezza temporanee utilizzate per effettuare la richiesta sono state emesse nel 2020. Questa policy può essere aggiornata ogni giorno a livello di codice per garantire che i membri dell'account utilizzino nuove credenziali.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::account-id:user/*",
    "Condition": {"DateGreaterThan": {"aws:TokenIssueTime": "2020-01-01T00:00:01Z"}}
  }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. La chiave `aws:TokenIssueTime` è presente nel contesto della richiesta solo quando il principale utilizza le credenziali temporanee per effettuare la richiesta. La chiave non è presente nelle AWS CLI richieste AWS API o AWS SDK effettuate utilizzando le chiavi di accesso. In questo esempio, se un utente IAM prova a visualizzare o modificare una chiave di accesso, la richiesta viene rifiutata.

Puoi utilizzare una variabile di policy con l'operatore di condizione `Date`.

Operatori di condizione booleani

Le condizioni booleane consentono di creare elementi `Condition` che limitano l'accesso in base al confronto con una chiave con valore "vero" o "falso".

Operatore di condizione	Descrizione
`Bool`	Corrispondenza booleana

Ad esempio, questa policy basata sull'identità utilizza l'operatore di condizione `Bool` con la chiave `aws:SecureTransport` per negare la replica degli oggetti e dei tag degli oggetti nel bucket di destinazione e i rispettivi contenuti se la richiesta non è su SSL.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BooleanExample",
      "Action": "s3:ReplicateObject",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    }
  ]
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. Il contesto della richiesta `aws:SecureTransport` restituisce vero o faso.

Puoi utilizzare una variabile di policy con l'operatore di condizione `Boolean`.

Operatori di condizione binari

L'operatore di condizione `BinaryEquals` consente di creare elementi `Condition` che testano i valori di chiave in formato binario. Viene effettuato un confronto del valore del byte di chiave specificato per il byte con una rappresentazione codificata in base 64 nella policy.

```
"Condition" : {
  "BinaryEquals": {
    "key" : "QmluYXJ5VmFsdWVJbkJhc2U2NA=="
  }
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono.

Puoi utilizzare una variabile di policy con l'operatore di condizione `Binary`.

Operatori di condizione con indirizzo IP

Gli operatori di condizione con indirizzo IP consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un indirizzo IPv4 o IPv6 o un intervallo di indirizzi IP. È possibile utilizzare questi operatori con la chiave `aws:SourceIp`. Il valore deve essere nel formato CIDR standard (ad esempio, 203.0.113.0/24 or 2001:DB8:1234:5678::/64). Se si specifica un indirizzo IP senza il prefisso di instradamento associato, IAM utilizza il valore predefinito di prefisso `/32`.

Alcuni AWS servizi supportano IPv6, utilizzando:: per rappresentare un intervallo di 0. Per sapere se un servizio supporta IPv6, consultare la documentazione per tale servizio.

Operatore di condizione	Descrizione
`IpAddress`	L'indirizzo o l'intervallo IP specificato
`NotIpAddress`	Tutti gli indirizzi IP tranne l'indirizzo o l'intervallo IP specificato

Ad esempio, la seguente istruzione utilizza l'operatore di condizione `IpAddress` con la chiave `aws:SourceIp` per specificare che la richiesta deve provenire dall'intervallo IP da 203.0.113.0 a 203.0.113.255.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::account-id:user/*",
    "Condition": {"IpAddress": {"aws:SourceIp": "203.0.113.0/24"}}
```

```
  }
}
```

La chiave di condizione `aws:SourceIp` risolve l'indirizzo IP da cui ha origine la richiesta. Se le richieste provengono da un'istanza Amazon EC2, `aws:SourceIp` viene valutato con l'indirizzo IP pubblico dell'istanza.

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. La chiave `aws:SourceIp` è sempre presente nel contesto della richiesta, tranne quando il richiedente utilizza un endpoint VPC per effettuare la richiesta. In questo caso, la condizione restituisce `false` e la richiesta è implicitamente rifiutata da questa istruzione.

Puoi utilizzare una variabile di policy con l'operatore di condizione `IpAddress`.

Nella policy di bucket di esempio riportata di seguito viene mostrato come combinare gli indirizzi IPv4 e IPv6 per coprire tutti gli indirizzi IP validi dell'organizzazione. Ti consigliamo di aggiornare le policy dell'organizzazione con gli intervalli di indirizzi IPv6 in aggiunta agli intervalli IPv4 di cui già disponi per garantire che le policy continuino a funzionare mentre effettui la transizione a IPv6.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "someservice:*",
    "Resource": "*",
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "203.0.113.0/24",
          "2001:DB8:1234:5678::/64"
        ]
      }
    }
  }
}
```

La chiave di condizione `aws:SourceIp` funziona solo in una policy JSON se si chiama l'API testata direttamente come utente. Se si utilizza invece un servizio per chiamare il servizio di destinazione per conto dell'utente, il servizio di destinazione visualizza l'indirizzo IP del servizio chiamante anziché l'indirizzo IP dell'utente originario. Questo può accadere, ad esempio, se chiami Amazon EC2 AWS CloudFormation per creare istanze per te. Attualmente non è disponibile alcun modo per passare

l'indirizzo IP di origine tramite un servizio di chiamata al servizio di destinazione per la valutazione in una policy JSON. Per questi tipi di chiamate API di servizi, non utilizzare la chiave di condizione `aws:SourceIp`.

Operatori di condizione con Amazon Resource Name (ARN)

Gli operatori di condizione con ARN (Amazon Resource Name) consentono di creare elementi `Condition` che limitano l'accesso in base al confronto di una chiave con un ARN. L'ARN è considerato una stringa.

Operatore di condizione	Descrizione
`ArnEquals` , `ArnLike`	Corrispondenza con distinzione maiuscole/minuscole dell'ARN. Ciascuno dei sei componenti delimitati da due punti dell'ARN viene verificato separatamente e ognuno di essi può includere più caratteri jolly (*) o un singolo carattere jolly (?) corrispondenti. Gli operatori di condizione `ArnEquals` e `ArnLike` si comportano allo stesso modo.
`ArnNotEquals` , `ArnNotLike`	Corrispondenza negativa per l'ARN. Gli operatori di condizione `ArnNotEquals` e `ArnNotLike` si comportano allo stesso modo.

Puoi utilizzare una variabile di policy con l'operatore di condizione `ARN`.

Nell'esempio di policy basata sulle risorse riportato di seguito viene illustrata una policy collegata a una coda Amazon SQS a cui si desidera inviare messaggi SNS. La policy autorizza Amazon SNS a inviare messaggi alla coda (o alle code) di propria scelta, ma solo se il servizio invia i messaggi per conto di un determinato argomento (o argomenti) di Amazon SNS. È possibile specificare la coda nel campo `Resource` e l'argomento Amazon SNS come valore per la chiave `SourceArn`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "123456789012"},
    "Action": "SQS:SendMessage",
    "Resource": "arn:aws:sqs:REGION:123456789012:QUEUE-ID",
    "Condition": {"ArnEquals": {"aws:SourceArn":
 "arn:aws:sns:REGION:123456789012:TOPIC-ID"}}
  }
```

```
}
```

Se la chiave specificata in una condizione di policy non è presente nel contesto della richiesta, i valori non corrispondono. La chiave `aws:SourceArn` è presente nel contesto della richiesta solo se una risorsa attiva un servizio per chiamare un altro servizio per conto del proprietario della risorsa. Se un utente IAM prova a eseguire direttamente questa operazione, la condizione restituisce `false` e la richiesta viene rifiutata implicitamente da questa istruzione.

... IfExists operatori di condizionamento

È possibile aggiungere `IfExists` alla fine di qualsiasi nome operatore di condizione ad eccezione della condizione `Null`, ad esempio `StringLikeIfExists`. Questa aggiunta ha lo scopo di dichiarare che "se la chiave di policy è presente nel contesto della richiesta, la chiave deve essere elaborata come specificato nella policy". Se la chiave non è presente, l'elemento della condizione viene valutato come "true". Altri elementi di condizione nell'istruzione possono comunque risultare in una mancata corrispondenza, ma non una chiave mancante se verificata tramite `...IfExists`. Se stai utilizzando un elemento `"Effect": "Deny"` con un operatore di condizione negato come `StringNotEqualsIfExists`, la richiesta viene comunque negata anche se manca il tag.

Esempio di utilizzo di **`IfExists`**

Molte chiavi di condizione descrivono informazioni su un determinato tipo di risorsa e sono presenti solo quando si accede a tale tipo di risorsa. Queste chiavi di condizione non sono presenti in altri tipi di risorse. Questo non causa problemi se l'istruzione della policy si applica a un solo tipo di risorsa. Tuttavia, esistono casi in cui una singola istruzione può essere applicata a più tipi di risorse, ad esempio quando l'istruzione della policy fa riferimento a operazioni di più servizi o quando una determinata operazione all'interno di un servizio accede a diversi tipi di risorse all'interno dello stesso servizio. In questi casi, l'inclusione di una chiave di condizione che si applica solo a una delle risorse nell'istruzione della policy può causare un errore dell'elemento `Condition` nell'istruzione della policy in modo tale che l'elemento `"Effect"` dell'istruzione non si applichi.

Ad esempio, considerare il seguente esempio di policy:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "THISPOLICYDOESNOTWORK",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
```

```
    "Condition": {"StringLike": {"ec2:InstanceType": [
      "t1.*",
      "t2.*",
      "m3.*"
    ]}}
  }
}
```

Lo scopo della policy precedente è di consentire all'utente di avviare qualsiasi istanza di tipo `t1`, `t2` o `m3`. Tuttavia, l'avvio di un'istanza richiede l'accesso a molte risorse oltre all'istanza stessa; ad esempio, immagini, coppie di chiavi, gruppi di sicurezza e così via. L'intera istruzione viene valutata rispetto a ogni risorsa necessaria per avviare l'istanza. Queste risorse aggiuntive non includono la chiave di condizione `ec2:InstanceType`, pertanto il controllo `StringLike` ha esito negativo e all'utente non è concessa la possibilità di avviare nessun tipo di istanza.

Per risolvere questo problema, utilizzare l'operatore di condizione `StringLikeIfExists`. In questo modo, il test viene effettuato solo se la chiave di condizione esiste. È possibile leggere la policy seguente come: "Se la risorsa da verificare include una chiave di condizione "`ec2:InstanceType`", permettere l'operazione solo se il valore della chiave inizia con `t1.`, `t2.` o `m3.`. Se la risorsa verificata non include quella chiave di condizione, non preoccuparti". L'asterisco (*) nei valori della chiave della condizione, se utilizzato con l'operatore di condizione `StringLikeIfExists`, viene interpretato come un jolly per ottenere corrispondenze parziali tra le stringhe. L'istruzione `DescribeActions` include le azioni necessarie per visualizzare l'istanza nella console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RunInstance",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "ec2:InstanceType": [
                        "t1.*",
                        "t2.*",
                        "m3.*"
             ]}}
        },
        {
```

```
            "Sid": "DescribeActions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        }]
}
```

Operatore di condizione per verificare la presenza di chiavi di condizione

Utilizza un operatore di condizione `Null` per verificare se una chiave di condizione è presente o meno al momento dell'autorizzazione. Nell'istruzione della policy, utilizza `true` (la chiave non esiste, è null) o `false` (la chiave esiste e il suo valore non è null).

Puoi utilizzare una variabile di policy con l'operatore di condizione `Null`.

Ad esempio, è possibile utilizzare questo operatore di condizione per determinare se un utente sta utilizzando per l'operazione le proprie credenziali o le credenziali temporanee. Se l'utente utilizza credenziali temporanee, la chiave `aws:TokenIssueTime` è presente e ha un valore. L'esempio seguente mostra una condizione che indica che l'utente non deve utilizzare credenziali temporanee (la chiave non deve esistere) affinché l'utente possa utilizzare l'API Amazon EC2.

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action":"ec2:*",
      "Effect":"Allow",
      "Resource":"*",
      "Condition":{"Null":{"aws:TokenIssueTime":"true"}}
  }
}
```

Condizioni con più chiavi di contesto o valori

Puoi utilizzare l'elemento `Condition` di una policy per testare più chiavi di contesto o valori per una singola chiave di contesto in una richiesta. Quando effettui una richiesta a AWS, a livello di

codice o tramite AWS Management Console, la richiesta include informazioni sul tuo principale, sull'operazione, sui tag e altro ancora. Puoi utilizzare le chiavi di contesto per testare i valori delle chiavi di contesto corrispondenti nella richiesta, con le chiavi di contesto specificate nella condizione della policy. Per ulteriori informazioni e i dati inclusi in una richiesta, consulta Il contesto della richiesta.

Argomenti

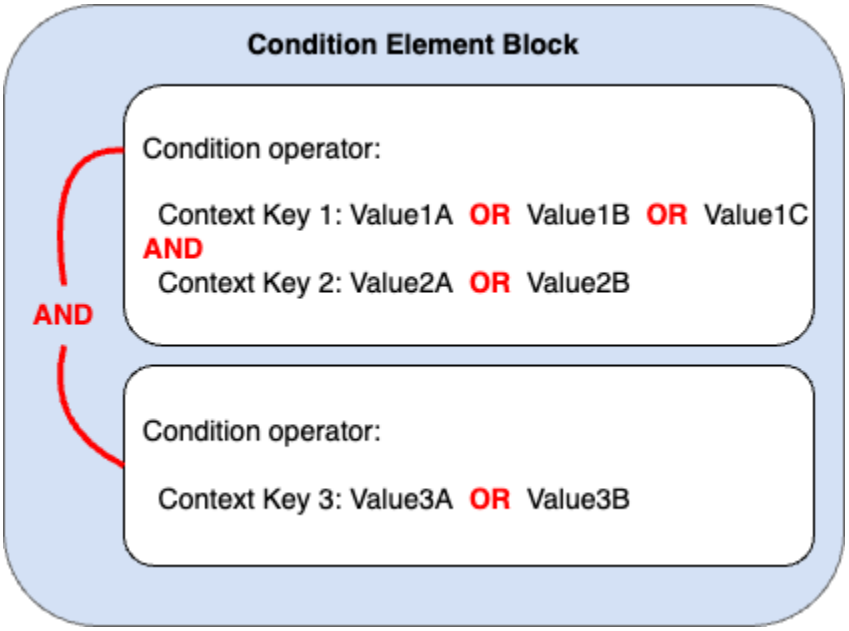
- Logica di valutazione per condizioni con più chiavi di contesto o valori
- Logica di valutazione per gli operatori di negazione della condizione di corrispondenza

Logica di valutazione per condizioni con più chiavi di contesto o valori

Un elemento `Condition` può contenere più operatori di condizione e ciascun operatore di condizione può includere a sua volta più coppie chiave-valore. La maggior parte delle chiavi di contesto supporta l'utilizzo di più valori, se non diversamente specificato.

- Se la tua policy contiene più operatori di condizione, questi vengono valutati utilizzando un `AND` logico.
- Se la policy contiene più chiavi di contesto collegate a un singolo operatore di condizione, le chiavi di contesto vengono valutate utilizzando un `AND` logico.
- Se un singolo operatore di condizione include valori multipli per una chiave di contesto, questi valori sono valutato con un `OR` logico.
- Se un singolo operatore di condizione di corrispondenza negata include valori multipli per una chiave di contesto, questi valori vengono valutati con un `NOR` logico.

Tutte le chiavi di contesto in un blocco di elementi condizionali devono essere risolte in true per richiamare il `Allow` desiderato o l'effetto `Deny`. La figura seguente illustra la logica di valutazione per una condizione con più operatori di condizione e coppie chiave-valore di contesto.



Ad esempio, la seguente policy del bucket S3 illustra come la figura precedente è rappresentata in una policy. Il blocco condizione utilizza operatori di condizione `StringEquals` e `ArnLike` e chiavi di contesto `aws:PrincipalTag` e`aws:PrincipalArn`. Per richiamare l'effetto `Allow` o `Deny` desiderato, tutte le chiavi di contesto nel blocco di condizione devono restituire il valore true. L'utente che effettua la richiesta deve avere entrambe le chiavi tag principali, dipartimento e ruolo, che includono uno dei valori chiave dei tag specificati nella policy. Inoltre, l'ARN principale dell'utente che effettua la richiesta deve corrispondere a uno dei valori `aws:PrincipalArn` specificati nella policy da valutare come true.

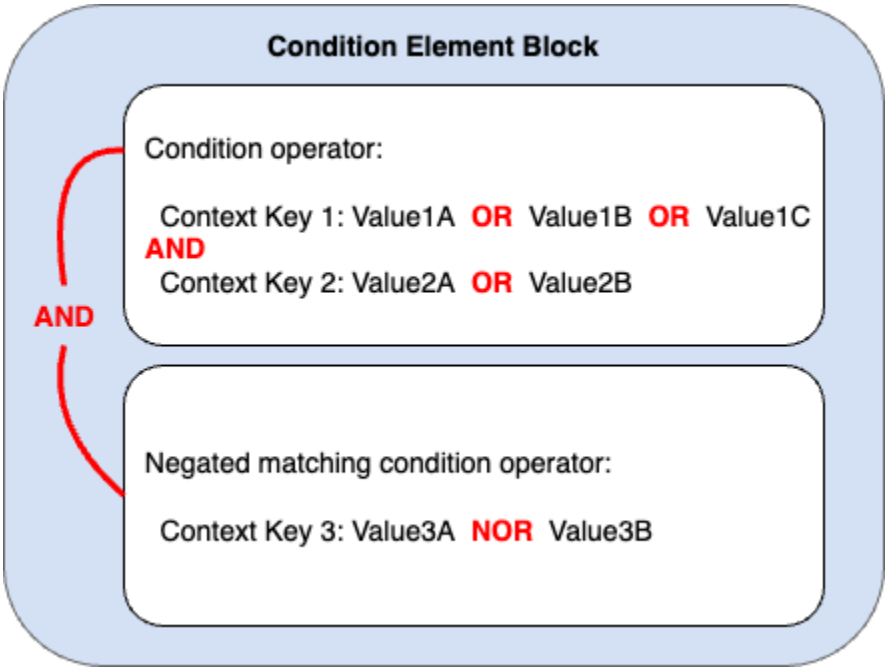
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222222222222:root"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": [
            "finance",
            "hr",
```

```
            "legal"
          ],
          "aws:PrincipalTag/role": [
            "audit",
            "security"
          ]
        },
        "ArnLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:user/Ana",
            "arn:aws:iam::222222222222:user/Mary"
          ]
        }
      }
    }
  ]
}
```

Logica di valutazione per gli operatori di negazione della condizione di corrispondenza

Alcuni operatori di condizione, ad esempio `StringNotEquals` o `ArnNotLike`, usano la corrispondenza negata per confrontare le coppie chiave-valore di contesto nella tua policy con le coppie chiave-valore di contesto in una richiesta. Quando più valori sono elencati in una singola chiave di contesto in una policy con operatori di negazione della condizione di corrispondenza le autorizzazioni efficaci funzionano come un `NOR` logico. Nella corrispondenza negata, un `NOR` o `NOT OR` restituisce true solo se tutti i valori restituiscono false.

La figura seguente illustra la logica di valutazione per una condizione con più operatori di condizione e coppie chiave-valore di contesto. La figura include un operatore di negazione della condizione di corrispondenza per la chiave di contesto 3.



Ad esempio, la seguente policy del bucket S3 illustra come la figura precedente è rappresentata in una policy. Il blocco condizione utilizza operatori di condizione `StringEquals` e `ArnNotLike` e chiavi di contesto `aws:PrincipalTag` e`aws:PrincipalArn`. Per richiamare l'effetto `Allow` o `Deny` desiderato, tutte le chiavi di contesto nel blocco di condizione devono restituire il valore true. L'utente che effettua la richiesta deve avere entrambe le chiavi tag principali, dipartimento e ruolo, che includono uno dei valori chiave dei tag specificati nella policy. Poiché l'operatore di condizione `ArnNotLike` utilizza la corrispondenza negata, l'ARN principale dell'utente che effettua la richiesta deve corrispondere a uno dei valori `aws:PrincipalArn` specificati nella policy da valutare come true.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222222222222:root"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": [
```

```
            "finance",
            "hr",
            "legal"
          ],
          "aws:PrincipalTag/role": [
            "audit",
            "security"
          ]
        },
        "ArnNotLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:user/Ana",
            "arn:aws:iam::222222222222:user/Mary"
          ]
        }
      }
    }
  ]
}
```

Chiavi di contesto a valore singolo vs multivalore

La differenza tra le chiavi di contesto a valore singolo e multivalore dipende dal numero di valori nel contesto della richiesta e non dal numero di valori nella condizione della policy.

- Le chiavi del contesto con condizione a valore singolo hanno al massimo un valore nel contesto della richiesta. Ad esempio, puoi applicare tag alle risorse in AWS. I tag delle risorse sono archiviati come coppie chiave-valore di tag. Una chiave di tag di risorsa può avere un singolo valore di tag. Pertanto, the section called “ResourceTag” è una chiave di contesto a valore singolo. Non utilizzare operatori con una chiave di contesto a valore singolo.
- Le chiavi di contesto con condizione multivalore possono avere più di un valore nel contesto della richiesta. Ad esempio, puoi taggare le risorse AWS e includere più coppie chiave-valore di tag in una richiesta. Pertanto, the section called “TagKeys” è una chiave di contesto multivalore. Le chiavi di contesto multivalore richiedono un operatore di condizione.

> **Important**
>
> Le chiavi di contesto multivalore richiedono un operatore di condizione. Non utilizzare operatori di condizione `ForAllValues` o `ForAnyValue` chiavi di contesto a valore singolo. Per ulteriori informazioni sugli operatori di condizione, vedereChiavi di contesto multivalore.

Le classificazione valore singolo e multivalore sono incluse nella descrizione di ciascuna chiave di contesto della condizione come tipo di valore nell'AWS chiavi di contesto della condizione globaleargomento. Il riferimento per l'autorizzazione del servizio utilizza una diversa classificazione dei tipi di valore per le chiavi di contesto multivalore nel seguente formato: un `ArrayOf` prefisso seguito dal tipo di categoria dell'operatore di condizione. Ad esempio `ArrayOfString` o `ArrayOfARN`.

Ad esempio, una richiesta può provenire al massimo da un endpoint VPC, quindi the section called "SourceVpce" è una chiave di contesto a valore singolo. Poiché un servizio può avere più di un nome del principale di servizio appartenente al servizio, leggi: PrincipalService NamesList è una chiave di contesto multivalore.

Qualsiasi chiave di contesto a valore singolo disponibile può essere utilizzata come policy variabile. Non è possibile utilizzare una chiave di contesto multivalore come policy variabile. Per ulteriori informazioni sulle variabili di policy, consultare Elementi delle policy IAM: variabili e tag.

Le chiavi di contesto multivalore richiedono gli operatori di condizione `ForAllValues` o `ForAnyValue`. Le chiavi di contesto che includono coppie chiave-valore come the section called "RequestTag" e the section called "ResourceTag" possono causare confusione perché possono essere presenti più valori *`tag-key`*. Dal momento che ogni *`tag-key`* può avere solo un valore, `aws:RequestTag` e `aws:ResourceTag`sono entrambe chiavi di contesto a valore singolo. L'utilizzo di operatori di set di condizioni con chiavi di contesto a valore singolo può portare a policy eccessivamente permissive.

Chiavi di contesto multivalore

Per confrontare la chiave di contesto della condizione con una chiave di contesto di richiesta con più valori chiave, devi utilizzare gli operatori di insiemi `ForAllValues` o `ForAnyValue`. Questi operatori di insieme sono usati per paragonare due insiemi di valori, ad esempio il set di tag in una richiesta e il set di tag in una condizione della policy.

I qualificatori `ForAllValues` e `ForAnyValue` aggiungono funzionalità di operazione di insieme all'operatore di condizione, in modo che tu possa testare chiavi di contesto della richiesta multivalore con più chiavi di contesto multiple in una condizione della policy. Inoltre, se includi una chiave di contesto di stringa multivalore nella policy con un carattere jolly o una variabile, devi utilizzare anche l'operatore di condizione `StringLike`. I valori multipli delle chiavi di condizione devono essere racchiusi tra parentesi quadre come in un array. Ad esempio, `"Key2":["Value2A", "Value2B"]`.

- `ForAllValues` - Questo qualificatore verifica il valore di ogni membro del set di richieste è un sottoinsieme del set di chiavi di contesto della condizione. La condizione restituisce true se ogni valore della chiave di contesto nella richiesta corrisponde ad almeno un valore nella policy. Restituisce true anche se non ci sono chiavi di contesto nella richiesta o se i valori delle chiavi si riducono a un set di dati nullo, ad esempio una stringa vuota. Per evitare che le chiavi di contesto mancanti o le chiavi di contesto con valori vuoti vengano valutate come vere, è possibile includere `Null` l'operatore di condizioni nella tua policy con un valore falso per verificare se la chiave di contesto esiste e il suo valore non è nullo.

  > **Important**
  >
  > Fai attenzione se usi `ForAllValues` con un effetto `Allow` perché ciò può essere eccessivamente permissivo se la presenza di chiavi di contesto mancanti o di chiavi di contesto con valori vuoti nel contesto della richiesta è imprevista. Puoi includere la condizione del `Null`operatore di condizioni nella tua policy con un valore falso per verificare se la chiave di contesto esiste e il suo valore non è nullo. Per vedere un esempio, consulta Controllo dell'accesso in base alle chiavi di tag.

- `ForAnyValue` - Questo test di qualificazione verifica se almeno un membro del set di valori di richiesta è corrispondente ad almeno un membro del set di valori delle chiavi di condizione della policy. La condizione restituisce true se uno qualsiasi dei valori della chiave di contesto corrisponde a un valore qualsiasi della chiave di contesto nella policy. Se non vi è una chiave di contesto corrispondente o di un set di dati vuoto, la condizione restituisce il valore false.

> **Note**
>
> La differenza tra le chiavi di contesto della condizione a valore singolo e multivalore dipende dal numero di valori nel contesto della richiesta e non dal numero di valori nella condizione della policy.

Esempi di policy delle condizioni

Nelle policy IAM, puoi specificare più valori per chiavi di contesto sia a valore singolo che multivalore per il confronto con il contesto della richiesta. La seguente serie di esempi di policy mostra le condizioni delle policy con più chiavi e valori di contesto.

> **Note**
>
> Per inviare una policy e includerla in questa guida di riferimento, utilizza il pulsante Feedback in fondo a questa pagina. Per esempi di policy basate su identità IAM, consulta Esempi di policy basate su identità IAM.

Esempi di policy relativi alle condizioni: chiavi di contesto a valore singolo

- Più blocchi di condizioni con chiavi di contesto a valore singolo. (Visualizza questo esempio.)
- Un blocco di condizioni con più chiavi e valori di contesto a valore singolo. (Visualizza questo esempio.)

Esempi di policy relativi alle condizioni: chiavi di contesto multivalore

- Policy di negazione con operatore del set di condizione `ForAllValues`. (Visualizza questo esempio.)
- Policy di negazione con operatore del set di condizione `ForAnyValue`. (Visualizza questo esempio.)

Esempi chiave di contesto multivalore

La seguente serie di esempi di policy mostra come creare condizioni politicy con chiavi di contesto multivalore.

Esempio: politica di rifiuto con operatore di set di condizioni ForAllValues

L'esempio seguente di policy basata sull'identità nega l'uso di azioni di tagging IAM quando nella richiesta sono inclusi prefissi specifici della chiave di tag. Ogni valore per la chiave di contesto `aws:TagKeys` include un carattere jolly (*) per la corrispondenza parziale delle stringhe. La policy include l'`ForAllValues`imposta l'operatore con la chiave di contesto `aws:TagKeys`perché la chiave di contesto della richiesta può includere più valori. La

condizione restituisce `aws:TagKeys` true se ogni valore nella richiesta corrisponde ad almeno un valore nella policy.

Quando si utilizza l'operatore di gruppo `ForAllValues`, questo restituisce true se non ci sono chiavi nella richiesta o se i valori delle chiavi si riducono a un set di dati nullo, ad esempio una stringa vuota. Per evitare che le chiavi di contesto mancanti o le chiavi di contesto con valori vuoti vengano valutate come vere, includi `Null` l'operatore di condizioni nella tua policy con un valore di `false` per verificare se la chiave di contesto nella richiesta esiste e il suo valore non è nullo.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRestrictedTags",
      "Effect": "Deny",
      "Action": [
        "iam:Tag*",
        "iam:UnTag*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Null": {
          "aws:TagKeys": "false"
        },
        "ForAllValues:StringLike": {
          "aws:TagKeys": [
            "key1*",
            "key2*",
            "key3*"
          ]
        }
      }
    }
  ]
```

```
}
```

Esempio: politica di negazione con operatore di set di condizioni ForAnyValue

Il seguente esempio di policy basata sull'identità nega la creazione di istantanee dei volumi di istanza EC2 se alcune istantanee sono contrassegnate con una delle chiavi di tag specificate nella policy,`environmentowebserver`. La policy include l'`ForAnyValue`operatore di insieme con la chiave di contesto `aws:TagKeys` perché la chiave di contesto della richiesta può includere più valori. Se la richiesta di etichettatura include uno dei valori chiave dei tag specificati nella policy, la `aws:TagKeys` chiave di contesto restituisce true richiamando l'effetto della policy di negazione.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza questa policy in combinazione con altre policy che consentono operazioni specifiche.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "arn:aws:ec2:us-west-2::snapshot/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": ["environment", "webserver"]
                }
            }
        }
    ]
}
```

Esempi di policy della chiave di contesto a valore singolo

La seguente serie di esempi di policy mostra come creare condizioni nella policy con chiavi di contesto a valore singolo.

Esempio: più blocchi di condizioni con chiavi di contesto a valore singolo

Quando un blocco di condizioni contiene più condizioni, ognuna con una singola chiave di contesto, tutte le chiavi di contesto devono risolversi in true per l'effetto `Allow` o `Deny` che si desidera richiamare. Quando si utilizzano operatori per la condizione di corrispondenza negata, la logica di valutazione del valore della condizione viene invertita.

Il seguente esempio consente agli utenti di creare volumi EC2 e applicare tag a tali volumi durante la creazione del volume. Il contesto della richiesta deve includere un valore per la chiave di contesto `aws:RequestTag/project` e il valore della chiave di contesto `aws:ResourceTag/environment` può essere qualsiasi cosa tranne la produzione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVolume",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:::volume/*",
      "Condition": {
        "StringLike": {
          "aws:RequestTag/project": "*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:region:account:*/*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceTag/environment": "production"
        }
      }
    }
  ]
}
```

Il contesto della richiesta deve includere un tag-valore del progetto e non può essere creato affinché una risorsa di produzione richiami l'effetto `Allow`. Il seguente volume EC2 è stato creato correttamente perché il nome del progetto è `Feature3` con un tag della risorsa `QA`.

```
aws ec2 create-volume \
    --availability-zone us-east-1a \
    --volume-type gp2 \
    --size 80 \
    --tag-specifications 'ResourceType=volume,Tags=[{Key=project,Value=Feature3},
{Key=environment,Value=QA}]'
```

Esempio: un blocco di condizioni con più chiavi di contesto a valore singolo

Quando un blocco di condizioni contiene più chiavi di contesto e ogni chiave di contesto ha valori multipli, ogni chiave di contesto deve risolversi in true in almeno un valore chiave per l'effetto `Allow` o `Deny` che si desidera richiamare. Quando si utilizzano operatori per la condizione di corrispondenza negata, la logica di valutazione del valore della chiave di contesto viene invertita.

L'esempio seguente consente agli utenti di avviare ed eseguire attività sui cluster Amazon Elastic Container Service.

- Il contesto della richiesta deve includere`production` O `pre-prod` per la `aws:RequestTag/environment` chiave di contesto E.
- La chiave di contesto `ecs:cluster` assicura che le attività vengano eseguite su entrambi i cluster `default1` O `default2` ARN ECS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:RunTask",
        "ecs:StartTask"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
```

```
          "aws:RequestTag/environment": [
            "production",
            "prod-backup"
          ]
        },
        "ArnEquals": {
          "ecs:cluster": [
            "arn:aws:ecs:us-east-1:111122223333:cluster/default1",
            "arn:aws:ecs:us-east-1:111122223333:cluster/default2"
          ]
        }
      }
    }
  ]
}
```

Elementi delle policy IAM: variabili e tag

Usa le variabili di policy AWS Identity and Access Management (IAM) come segnaposto quando non conosci il valore esatto di una risorsa o di una chiave di condizione quando scrivi la policy.

> **Note**
>
> Se AWS non riesci a risolvere una variabile, l'intera istruzione potrebbe non essere valida. Ad esempio, se utilizzi la variabile `aws:TokenIssueTime`, questa viene risolta in un valore solo quando il richiedente è stato autenticato tramite le credenziali temporanee (un ruolo IAM). Per evitare che le variabili causino istruzioni non valide, usa il... IfExists operatore di condizione.

Argomenti

Introduzione

Nelle policy IAM, numerose operazioni consentono di assegnare un nome a risorse specifiche per le quali si desidera controllare l'accesso. Ad esempio, la policy di seguito consente all'utente di elencare, leggere e scrivere gli oggetti nel bucket S3 `DOC-EXAMPLE-BUCKET` per i progetti `marketing`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
      "Condition": {"StringLike": {"s3:prefix": ["marketing/*"]}}
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/marketing/*"]
    }
  ]
}
```

In alcuni casi, potresti non conoscere il nome esatto della risorsa al momento di scrivere la policy. Puoi generalizzare la policy per renderla utilizzabile da molti utenti senza bisogno di effettuare una copia univoca per ciascun utente. Invece di creare una policy separata per ciascun utente, consigliamo di creare una singola policy di gruppo che funzioni per tutti gli utenti che appartengono a tale gruppo.

Utilizzo delle variabili nelle policy

È possibile definire valori dinamici all'interno delle policy utilizzando variabili di policy che impostano i segnaposti in una policy.

Le variabili sono contrassegnate utilizzando un prefisso **$**seguito da una coppia di parentesi graffe (**{ }**) che includono il nome della variabile del valore della richiesta.

Quando la policy viene valutata, le variabili di policy vengono sostituite con valori provenienti dalle chiavi di contesto condizionali inoltrate nella richiesta. Le variabili possono essere utilizzate nelle policy basate sull'identità, nelle policy delle risorse, nelle policy di controllo dei servizi, nelle policy di sessione e nelle policy degli endpoint VPC. Anche le policy basate sull'identità utilizzate come limiti delle autorizzazioni supportano le variabili di policy.

Le chiavi di contesto delle condizioni globali possono essere utilizzate come variabili nelle richieste tra AWS i servizi. Anche le chiavi di condizione specifiche del servizio possono essere utilizzate come variabili quando interagiscono con le risorse AWS , ma sono disponibili soltanto quando le richieste vengono effettuate su risorse che le supportano. Per un elenco delle chiavi di contesto disponibili per ogni AWS servizio e risorsa, consulta il Service Authorization Reference. In determinate circostanze, non è possibile inserire un valore nelle chiavi di contesto delle condizioni globali. Per ulteriori informazioni su ciascuna chiave, consulta la pagina AWS Chiavi di contesto delle condizioni globali .

> **Important**
>
> - I nomi delle chiavi non fanno distinzione tra maiuscole e minuscole. Ad esempio, `aws:CurrentTime` è uguale a `AWS:currenttime`.
> - È possibile utilizzare qualsiasi chiave di condizione a valore singolo come variabile. Non è possibile utilizzare una chiave della condizione multi-valore come variabile.

L'esempio seguente mostra una policy per un utente o un ruolo IAM che sostituisce il nome di una risorsa specifica con una variabile di policy. Puoi riutilizzare questa policy sfruttando i vantaggi della chiave di condizione `aws:PrincipalTag`. Quando questa policy viene valutata, `${aws:PrincipalTag/team}` consente le operazioni solo se il nome del bucket termina con il nome del team dal tag del principale `team`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
      "Condition": {"StringLike": {"s3:prefix": ["${aws:PrincipalTag/team}/*"]}}
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/${aws:PrincipalTag/team}/*"]
    }
  ]
}
```

La variabile viene contrassegnata utilizzando un prefisso $ seguito da una coppia di parentesi graffe ({ }). All'interno dei caratteri ${ } è possibile includere il nome del valore ricavato dalla richiesta da utilizzare nella policy. I valori che possono essere utilizzati sono descritti più avanti in questa pagina.

Per maggiori dettagli su questa chiave di condizione globale, consulta aws:PrincipalTag//tag-key nell'elenco di chiavi di condizioni globali.

> **Note**
>
> Per usare le variabili di policy, è necessario che l'elemento `Version` sia incluso in una dichiarazione. Inoltre, la versione deve essere impostata su una versione che supporti le variabili di policy. Le variabili sono state introdotte a partire dalla versione `2012-10-17`. Le versioni precedenti del linguaggio di policy non supportano le variabili. Se l'elemento `Version` non viene incluso e impostato su una data appropriata, alcune variabili, come ad esempio `${aws:username}`, saranno trattate come stringhe letterali nella policy.
> Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, viene creata una versione della policy quando si modifica una policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare the section called “Version”. Per ulteriori informazioni sulle versioni di policy, consultare the section called “Controllo delle versioni delle policy IAM”.

Una policy che consente a un principale di ottenere oggetti dal percorso /David di un bucket S3 è simile alla seguente:

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [{
    "Effect": "Allow",
    "Action": ["s3:GetObject"],
    "Resource": ["arn:aws:s3::DOC-EXAMPLE-BUCKET/David/*"]
  }]
}
```

Se questa policy è collegata all'utente `David`, tale utente ottiene gli oggetti dal proprio bucket S3, ma potrebbe essere necessario creare una policy separata per ogni utente che include il nome dell'utente. Ciascuna policy dovrà quindi essere collegata ai singoli utenti.

Utilizzando una variabile di policy, è possibile creare policy che possono essere riutilizzate. La seguente policy consente a un utente di ottenere oggetti da un bucket Amazon S3 se il valore tag-chiave per `aws:PrincipalTag` corrisponde al valore del `owner` tag-chiave inviato nella richiesta.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowUnlessOwnedBySomeoneElse",
    "Effect": "Allow",
    "Action": ["s3:GetObject"],
    "Resource": ["*"],
    "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/owner": "${aws:PrincipalTag/owner}"
        }
      }
    }
  ]
}
```

Se utilizzi una variabile di policy al posto di un utente di questo tipo, non è necessaria una policy separata per ogni singolo utente. Nell'esempio seguente, la policy è collegata a un ruolo IAM assunto dai Product Manager tramite credenziali di sicurezza temporanee. Quando un utente richiede di aggiungere un oggetto Amazon S3, IAM sostituisce il valore del tag `dept` della richiesta corrente per la variabile `${aws:PrincipalTag}` e valuta la policy.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowOnlyDeptS3Prefix",
```

```
    "Effect": "Allow",
    "Action": ["s3:GetObject"],
    "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/${aws:PrincipalTag/dept}/*"],
    }
  ]
}
```

Tag come variabili di policy

In alcuni AWS servizi è possibile associare attributi personalizzati alle risorse create da tali servizi. Ad esempio, puoi applicare tag a bucket Amazon S3 o a utenti IAM. Questi tag sono coppie chiave-valore. È possibile definire il nome della chiave di tag e il valore associato al nome della chiave. Ad esempio, puoi creare un tag con una chiave **`department`** e un valore **`Human Resources`**. Per ulteriori informazioni sul tagging delle entità IAM, consulta Tagging delle risorse IAM. Per informazioni sul tagging delle risorse create da altri servizi AWS , consulta la documentazione di tali servizi. Per ulteriori informazioni sull'utilizzo dell'editor di tag, consulta l'articolo relativo all'utilizzo dell'editor di tag nella Guida per l'utente della AWS Management Console .

Puoi applicare tag alle risorse IAM per semplificare il rilevamento, l'organizzazione e il monitoraggio delle risorse IAM. Puoi inoltre applicare tag alle identità IAM per controllare l'accesso alle risorse o al tagging. Per ulteriori informazioni sull'uso dei tag per controllare l'accesso, consulta Controllo dell'accesso a e per utenti e ruoli IAM mediante i tag.

Casi in cui è possibile utilizzare le variabili di policy

Le variabili di policy possono essere utilizzate nell'elemento `Resource` e per confrontare stringhe nell'elemento `Condition`.

Elemento risorsa

È possibile utilizzare una variabile criterio nell'elemento `Resource`, ma solo nella parte risorsa dell'ARN. Questa parte dell'ARN appare dopo i quinti due punti (:). Non è possibile utilizzare una variabile per sostituire parti dell'ARN prima dei quinti due punti, ad esempio il servizio o l'account. Per ulteriori informazioni sul formato ARN, consulta ARN IAM.

Per sostituire una parte di un ARN con un valore di tag, inserisci il prefisso e il nome della chiave in `${ }`. Ad esempio, il seguente elemento Risorsa si riferisce solo a un bucket con lo stesso nome del valore nel tag department dell'utente richiedente.

```
"Resource": ["arn:aws::s3:::bucket/${aws:PrincipalTag/department}"]
```

Molte AWS risorse utilizzano ARN che contengono un nome creato dall'utente. La seguente policy IAM garantisce che solo gli utenti voluti con i valori di tag access-project, access-application e access-environment corrispondenti possono modificare le relative risorse. Inoltre, utilizzando le corrispondenze con carattere jolly *, sono in grado di consentire suffissi personalizzati per i nomi delle risorse.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessBasedOnArnMatching",
      "Effect": "Allow",
      "Action": [
        "sns:CreateTopic",
        "sns:DeleteTopic"],
      "Resource": ["arn:aws:sns:*:*:${aws:PrincipalTag/access-project}-
${aws:PrincipalTag/access-application}-${aws:PrincipalTag/access-environment}-*"
      ]
    }
  ]
}
```

Elemento condizione

È possibile utilizzare una variabile di criterio per i valori `Condition` in qualsiasi condizione che coinvolga gli operatori stringa o gli operatori ARN. Gli operatori stringa includono `StringEquals`, `StringLike` e `StringNotLike`. Gli operatori ARN includono `ArnEquals` e `ArnLike`. Non è possibile utilizzare una variabile di criterio con altri operatori, ad esempio `Numeric`, `Date`, `Boolean`, `Binary`, `IP Address` o `Null`. Per ulteriori informazioni sugli operatori delle condizioni, vedere Elementi della policy JSON IAM: operatori di condizione.

Quando fai riferimento a un tag in un'espressione dell'elemento `Condition`, utilizza il prefisso e il nome della chiave pertinenti come chiave di condizione. Quindi utilizza il valore che desideri testare nel valore della condizione.

Ad esempio, la seguente policy di esempio consente l'accesso completo agli utenti; ma solo se il tag `costCenter` è collegato all'utente. Il tag deve inotlre avere un valore pari a `12345` o `67890`. Se il tag non ha nessun valore o ha qualsiasi altro valore, la richiesta ha esito negativo.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iam:*user*"
       ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:ResourceTag/costCenter": [ "12345", "67890" ]
        }
      }
    }
  ]
}
```

Variabili di policy senza valore

Quando le variabili di policy fanno riferimento a una chiave di contesto delle condizioni che non ha valore o non è presente nel contesto di autorizzazione per una richiesta, il valore è effettivamente nullo. Non esiste un valore uguale o simile. Le chiavi di contesto delle condizioni potrebbero non essere presenti nel contesto di autorizzazione quando:

- Si utilizzano le chiavi di contesto delle condizioni specifiche del servizio nelle richieste inviate a risorse che non supportano tale chiave di condizione.
- I tag sulle sessioni, sulle risorse, sulle richieste o sui principali IAM non sono presenti.
- Altre circostanze, come specificate per ogni chiave di contesto delle condizioni globali alla pagina AWS chiavi di contesto della condizione globale.

Quando si utilizza una variabile senza valore nell'elemento della condizione di una policy IAM, gli Elementi della policy JSON IAM: operatori di condizione come `StringEquals` o `StringLike` non corrispondono e l'istruzione della policy non ha effetto.

Gli operatori di condizione invertiti come `StringNotEquals` o `StringNotLike` corrispondono a un valore nullo, poiché il valore della chiave di condizione che stanno verificando non corrisponde o non è uguale al valore effettivamente nullo.

Nel seguente esempio, per consentire l'accesso `aws:principaltag/Team` deve essere uguale a `s3:ExistingObjectTag/Team`. L'accesso è esplicitamente negato quando `aws:principaltag/Team` non è impostato. Se una variabile che non ha valore nel contesto di autorizzazione viene

utilizzata come parte dell'elemento `Resource` o `NotResource` di una policy, la risorsa che include una variabile di policy senza valore non corrisponderà ad alcuna risorsa.

```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Effect": "Deny",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::/example-bucket/*",
    "Condition": {
      "StringNotEquals": {
        "s3:ExistingObjectTag/Team": "${aws:PrincipalTag/Team}"
       }
      }
    }
  ]
}
```

Richiesta di informazioni utilizzabili per le variabili di policy

È possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi nel contesto della richiesta con i valori chiave specificati nella policy. Quando si utilizza una variabile di policy, AWS sostituisce un valore della chiave di contesto della richiesta al posto della variabile nella policy.

Valori della chiave dell'entità principale

I valori per `aws:username`, `aws:userid` e `aws:PrincipalType` dipendono dal tipo di principale che ha avviato la richiesta. Ad esempio, la richiesta può essere effettuata utilizzando le credenziali di un utente IAM, di un ruolo IAM o dell' Utente root dell'account AWS. La seguente lista mostra i valori di tali chiavi per i diversi tipi di principale.

- Utente root dell'account AWS
  - `aws:username`: (non presente)
  - `aws:userid`: Account AWS ID
  - `aws:PrincipalType`: `Account`
- Utente IAM
  - `aws:username`: *`nome-utente-IAM`*
  - `aws:userid`: ID univoco

  - `aws:PrincipalType`: `User`
- Utente federato
  - `aws:username`: (non presente)
  - `aws:userid`: *`account`*:*`nome-specificato-intermediario`*
  - `aws:PrincipalType`: `FederatedUser`
- Utente federato Web e utente federato SAML

> **Note**
>
> Per informazioni sulle chiavi di policy disponibili quando si utilizza la federazione OIDC, vedere. ???

  - `aws:username`: (non presente)
  - `aws:userid`: (non presente)
  - `aws:PrincipalType`: `AssumedRole`
- Ruolo presunto
  - `aws:username`: (non presente)
  - `aws:userid`: *`id-ruolo`*:*`nome-ruolo-specificato-intermediario`*
  - `aws:PrincipalType`: `Assumed role`
- Ruolo assegnato a un'istanza Amazon EC2
  - `aws:username`: (non presente)
  - `aws:userid`: *`id-ruolo`*:*`id-istanza-ec2`*
  - `aws:PrincipalType`: `Assumed role`
- Chiamante anonimo (solo Amazon SQS, Amazon SNS e Amazon S3)
  - `aws:username`: (non presente)
  - `aws:userid`: (non presente)
  - `aws:PrincipalType`: `Anonymous`

Per gli elementi di questo elenco, nota quanto segue:

- non presente significa che il valore non è riportato nelle informazioni della richiesta corrente e qualsiasi tentativo di trovare una corrispondenza avrà esito negativo e l'istruzione viene considerata non valida.
- *id-ruolo* è un identificatore univoco assegnato a ciascun ruolo al momento della creazione. È possibile visualizzare l'ID del ruolo con il AWS CLI comando: `aws iam get-role --role-name` *rolename*
- *nome-specificato-intermediario* e *nome-ruolo-specificato-intermediario* sono nomi che vengono trasmessi dal processo di richiamo (ad esempio un'applicazione o servizio) quando si effettua una chiamata per ottenere credenziali provvisorie.
- *ec2-instance-id* è un valore assegnato all'istanza all'avvio e viene visualizzato nella pagina Istanzedella console Amazon EC2. Puoi anche visualizzare l'ID dell'istanza eseguendo il AWS CLI comando: `aws ec2 describe-instances`

Informazioni disponibili nelle richieste per gli utenti federati

Gli utenti federati vengono autenticati utilizzando un sistema diverso da IAM. Ad esempio, un'azienda potrebbe disporre di un'applicazione da utilizzare internamente che effettua chiamate verso AWS. Fornire un'identità IAM a ogni utente dell'azienda che utilizza l'applicazione potrebbe risultare poco pratico. Al contrario, l'azienda può utilizzare un'applicazione proxy (livello intermedio) con una singola identità IAM, oppure un provider di identità (IdP) SAML. L'applicazione proxy o l'IdP SAML autentica i singoli utenti individuali tramite la rete aziendale. Un'applicazione proxy può quindi utilizzare la propria identità di IAM per ottenere credenziali di sicurezza provvisorie per i singoli utenti. Un IdP SAML può in effetti scambiare informazioni sull'identità AWS per credenziali di sicurezza temporanee. Le credenziali temporanee possono quindi essere utilizzate per accedere alle risorse. AWS

Allo stesso modo, potresti creare un'applicazione per dispositivi mobili che richiede l'accesso alle risorse AWS . In tal caso, puoi utilizzare la federazione OIDC, in cui l'app autentica l'utente utilizzando un provider di identità noto come Login with Amazon, Amazon Cognito, Facebook o Google. A questo punto, l'applicazione potrà utilizzare le informazioni di autenticazione dell'utente fornite da tali provider per ottenere le credenziali di sicurezza temporanee per l'accesso alle risorse AWS .

Il modo consigliato per utilizzare la federazione OIDC consiste nello sfruttare Amazon Cognito e AWS gli SDK per dispositivi mobili. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Guida per l'utente di Amazon Cognito
- Scenari comuni per le credenziali temporanee

Caratteri speciali

Alcune variabili di policy speciali e predefinite hanno valori fissi che consentono di rappresentare caratteri che altrimenti avrebbero un significato diverso. Se tali caratteri speciali fanno parte della stringa per cui stai cercando una corrispondenza e vengono inseriti letteralmente, il loro significato sarebbe frainteso. Ad esempio, se nella stringa inserisci un asterisco (*), questo non sarà interpretato come un semplice asterisco, ma come un carattere jolly corrispondente a tutti i caratteri. In tali casi, puoi utilizzare le seguenti variabili di policy predefinite:

- ${*} - da usare quando devi inserire un asterisco (*).
- ${?} - da usare quando devi inserire un punto interrogativo (?).
- ${$} - da usare quando devi inserire il simbolo del dollaro ($).

Queste variabili di policy predefinite possono essere inserite in qualsiasi stringa in cui è possibile utilizzare le normali variabili di policy.

Specifica dei valori di default

Per aggiungere un valore di default a una variabile, racchiudi il valore di default tra virgolette singole (`' '`) e separa il testo della variabile e il valore di default con una virgola e uno spazio (`, `).

Ad esempio, se a un principale è applicato un tag`team=yellow`, è possibile accedere al bucket Amazon S3 `ExampleCorp's` denominato `DOC-EXAMPLE-BUCKET-yellow`. Una policy con questa risorsa consente ai membri del team di accedere al bucket del team, ma non a quelli di altri team. Per gli utenti senza tag team, la policy imposta un valore di default di `company-wide` per il nome del bucket. Questi utenti possono accedere solo al bucket `DOC-EXAMPLE-BUCKET-company-wide` dove possono visualizzare informazioni generali, come le istruzioni per entrare a far parte di un team.

```
"Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET-${aws:PrincipalTag/team, 'company-wide'}"
```

Ulteriori informazioni

Per ulteriori informazioni sulle policy, consultare:

- Policy e autorizzazioni in IAM
- Esempi di policy basate su identità IAM
- Documentazione di riferimento degli elementi delle policy JSON IAM

- Logica di valutazione delle policy
- Federazione OIDC

Elementi della policy JSON IAM: tipi di dati supportati

Questa sezione elenca i tipi di dati che sono supportati quando si specificano valori in policy JSON. Il linguaggio della policy non supporta tutti i tipi per ciascun elemento della policy; per informazioni su ciascun elemento, consultare le sezioni precedenti.

- Stringhe
- Numeri (interi e valori a virgola mobile)
- Booleano
- Null
- Elenchi
- Mappe
- Strutture (sono solo mappe nidificate)

La tabella seguente associa ogni tipo di dati alla serializzazione. Notare che tutte le policy devono essere in UTF-8. Per informazioni sui tipi di dati JSON, consulta RFC 4627.

Type	JSON
Stringa	Stringa
Numero intero	Numero
Float	Numero
Booleano	true false
Null	null
Data	Stringa in linea con il profilo W3C di ISO 8601
IpAddress	Stringa in linea con RFC 4632
Elenco	Array

Type	JSON
Oggetto	Oggetto

Logica di valutazione delle policy

Quando un principale tenta di utilizzare l' AWS Management Console, l' AWS API o il AWS CLI, quel principale invia una richiesta a. AWS Quando un AWS servizio riceve la richiesta, AWS completa diversi passaggi per determinare se consentire o rifiutare la richiesta.

1. Autenticazione: autentica AWS innanzitutto il principale che effettua la richiesta, se necessario. Questo passaggio non è necessario per alcuni servizi, ad esempio Amazon S3, che consentono alcune richieste da parte di utenti anonimi.
2. Elaborazione del contesto della richiesta— AWS elabora le informazioni raccolte nella richiesta per determinare quali politiche si applicano alla richiesta.
3. Valutazione delle policy in un singolo account— AWS valuta tutti i tipi di policy, che influiscono sull'ordine in cui le politiche vengono valutate.
4. Determinazione se una richiesta è consentita o rifiutata in un account— elabora AWS quindi le politiche in base al contesto della richiesta per determinare se la richiesta è consentita o rifiutata.

Elaborazione del contesto della richiesta

AWS elabora la richiesta per raccogliere le seguenti informazioni in un contesto di richiesta:

- Azioni o operazioni: le azioni o le operazioni che il principale vuole eseguire.
- Risorse: l'oggetto AWS risorsa su cui vengono eseguite le azioni o le operazioni.
- Principale: l'utente, il ruolo, l'utente federato o l'applicazione che ha inviato la richiesta. Le informazioni sull'entità principale includono le policy associate a tale entità principale.
- Dati di ambiente: informazioni sull'indirizzo IP, l'agente utente, lo stato SSL abilitato o l'ora del giorno.
- Dati sulla risorsa – I dati correlati alla risorsa che viene richiesta. Possono essere incluse informazioni quali un nome di tabella di DynamoDB o un tag su un'istanza Amazon EC2.

AWS utilizza quindi queste informazioni per trovare le politiche che si applicano al contesto della richiesta.

Valutazione delle policy in un singolo account

Il modo in cui AWS valuta le politiche dipende dai tipi di politiche che si applicano al contesto della richiesta. I tipi di policy elencati di seguito in ordine di frequenza possono essere utilizzati in un singolo Account AWS. Per ulteriori informazioni su questi tipi di policy, consulta Policy e autorizzazioni in IAM. Per informazioni su come AWS valuta le politiche per l'accesso tra account diversi, consulta. Logica di valutazione della policy multiaccount

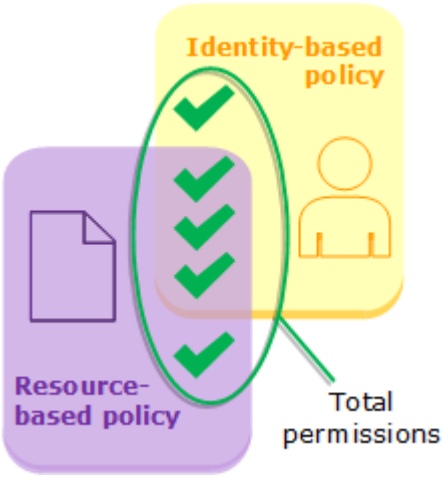
1. Policy basate su identità: le policy basate su identità sono collegate a un'identità IAM (utente, gruppo di utenti o ruolo) e concede le autorizzazioni per entità IAM (utenti e ruoli). Se a una richiesta si applicano solo le politiche basate sull'identità, verifica che tutte queste politiche ne AWS accertino almeno una. `Allow`
2. Policy basate sulle risorse - Le policy basate sulle risorse concedono autorizzazioni al principale (account, utente, ruolo e principali di sessione come sessioni di ruolo e utenti federati IAM) specificato come principale. Le autorizzazioni definiscono ciò che l'entità principale può fare con la risorsa a cui è collegata la policy. Se le politiche basate sulle risorse e le politiche basate sull'identità si applicano entrambe a una richiesta, controlla tutte le politiche per verificarne AWS almeno una. `Allow` Quando vengono valutate le policy sulle risorse, l'ARN del principale specificato nella policy determina se le negazioni implicite in altri tipi di policy sono applicabili alla decisione finale.
3. Limiti delle autorizzazioni IAM: i limiti delle autorizzazioni sono una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo). Quando si imposta un limite delle autorizzazioni per un'entità, l'entità può eseguire solo le operazioni consentite dalle sue policy basate su identità e dai suoi limiti delle autorizzazioni. In alcuni casi, un rifiuto implicito in un limite delle autorizzazioni può limitare le autorizzazioni concesse da una policy basata sulle risorse. Per ulteriori informazioni, consulta Determinazione se una richiesta è consentita o rifiutata in un account più avanti in questo argomento.
4. AWS Organizations policy di controllo dei servizi (SCP) — Organizations Gli SCP specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU). Il limite SCP si applica ai principali degli account dei membri, compresi ciascuno. Utente root dell'account AWS Se una SCP è presente, le policy basate su identità e le policy basate su risorse concedono autorizzazioni alle entità negli account dei membri solo se tali policy e l'SCP consentono l'operazione. Se sono presenti sia un limite delle autorizzazioni sia una SCP, il limite, l'SCP e la policy basata su identità devono tutti consentire l'operazione.

5. Policy di sessione: le policy di sessione sono policy avanzate che si inviano come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Per creare una sessione del ruolo in modo programmatico, è possibile utilizzare una delle operazioni API `AssumeRole*`. Quando esegui questa operazione e passi le policy di sessione, le autorizzazioni della sessione risultante sono l'intersezione della policy basata su identità dell'utente dell'entità IAM e delle policy di sessione. Per creare una sessione per l'utente federato, si utilizzano le chiavi di accesso dell'utente IAM per chiamare in modo programmatico l'operazione API `GetFederationToken`. Una policy basata sulle risorse ha un effetto diverso sulla valutazione delle autorizzazioni della policy di sessione. La differenza dipende dal fatto che l'ARN dell'utente o del ruolo o l'ARN della sessione sia elencato come il principale nella policy basata sulle risorse. Per ulteriori informazioni, consulta Policy di sessione.

Occorre ricordare che un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

Valutazione delle policy basate su identità con policy basate su risorse

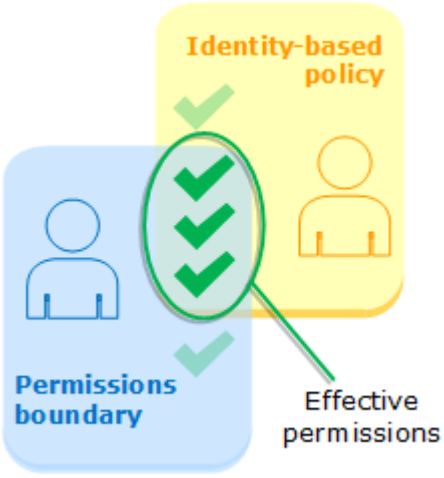
Le policy basate su identità e le policy basate su risorse concedono autorizzazioni alle identità o alle risorse a cui sono collegate. Quando un'entità IAM (utente o ruolo) richiede l'accesso a una risorsa all'interno dello stesso account, AWS valuta tutte le autorizzazioni concesse dalle politiche basate sull'identità e sulle risorse. Le autorizzazioni risultanti sono le autorizzazioni totali dei due tipi. Se un'azione è consentita da una policy basata sull'identità, una policy basata sulle risorse o entrambe, allora consente l'azione. AWS Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione.



Valutazione delle policy basate su identità con i limiti delle autorizzazioni

Quando si AWS valutano le politiche basate sull'identità e i limiti delle autorizzazioni per un utente, le autorizzazioni risultanti sono l'intersezione delle due categorie. Ciò significa che quando aggiungi un limite delle autorizzazioni a un utente con policy basate su identità esistenti, potresti ridurre il numero

di operazioni che l'utente può eseguire. Di contro, quando rimuovi un limite delle autorizzazioni da un utente, potresti aumentare il numero di operazioni che può eseguire. Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione. Per informazioni su come altri tipi di policy vengono valutati con i limiti delle autorizzazioni, consulta Valutazione delle autorizzazioni valide con i limiti.



Valutazione delle policy basate su identità con le SCP di Organizations

Quando un utente appartiene a un account che è un membro di un'organizzazione, le autorizzazioni risultanti sono l'intersezione delle policy dell'utente e dell'SCP. Ciò significa che un'operazione deve essere consentita sia dalla policy basata su identità sia dall'SCP. Un rifiuto esplicito in una di queste policy sostituisce l'autorizzazione.

Puoi scoprire se il tuo account è un membro di un'organizzazione in AWS Organizations. I membri dell'organizzazione potrebbero essere influenzati da una SCP. Per visualizzare questi dati utilizzando il AWS CLI comando o l'operazione AWS API, devi disporre delle autorizzazioni per l'`organizations:DescribeOrganization`azione per la tua entità Organizations. È necessario disporre delle autorizzazioni aggiuntive per eseguire l'operazione nella console Organizations. Per

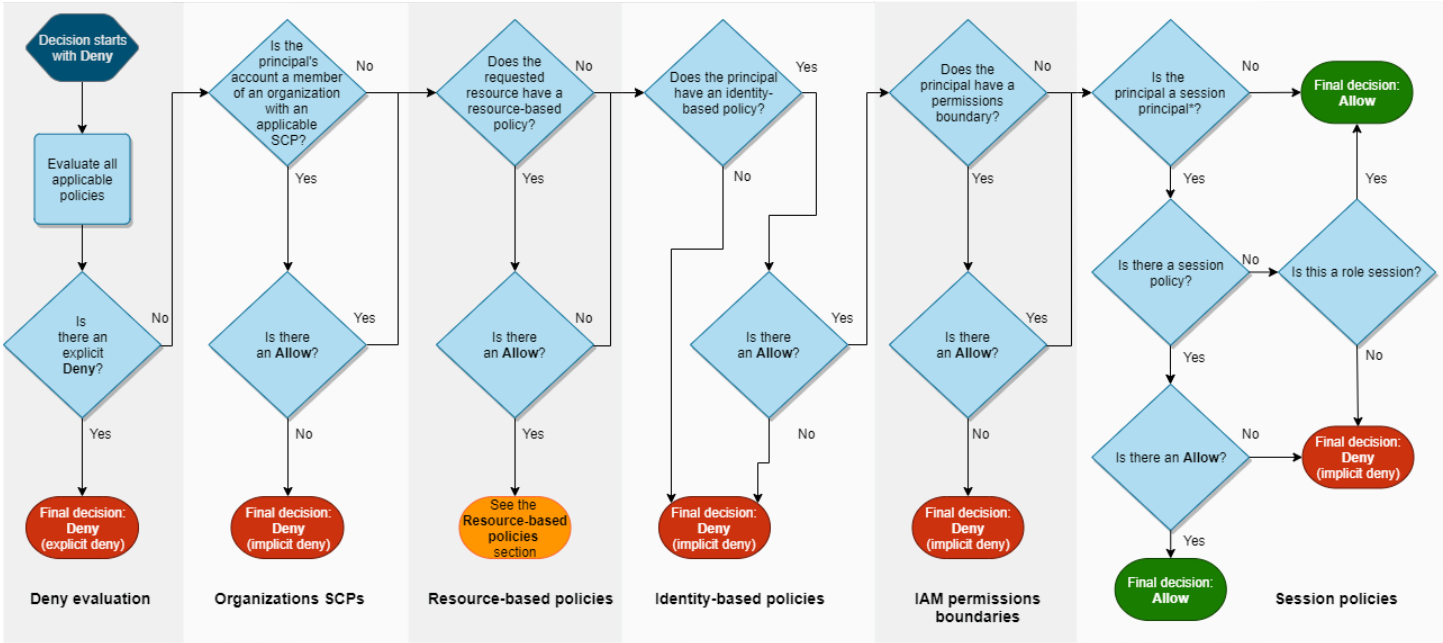
sapere se un SCP sta negando l'accesso a una richiesta specifica o per modificare le autorizzazioni effettive, contatta il tuo amministratore. AWS Organizations

Determinazione se una richiesta è consentita o rifiutata in un account

Supponiamo che un principale invii una richiesta AWS di accesso a una risorsa nello stesso account dell'entità del principale. Il codice di AWS applicazione decide se la richiesta deve essere accolta o rifiutata. AWS valuta tutte le politiche applicabili al contesto della richiesta. Di seguito è riportato un riepilogo della logica di AWS valutazione delle politiche all'interno di un singolo account.

- Per impostazione predefinita, tutte le richieste vengono negate implicitamente ad eccezione di Utente root dell'account AWS, che ha accesso completo.
- Un'autorizzazione esplicita in una policy basata su identità o basata su risorse sostituisce questa impostazione predefinita.
- Se è presente un limite delle autorizzazioni, una SCP di Organizations oppure una policy di sessione, potrebbe sovrascrivere l'autorizzazione con un rifiuto implicito.
- Un rifiuto esplicito in una policy sostituisce qualsiasi permesso.

Il seguente diagramma di flusso fornisce i dettagli su come la decisione viene presa. Questo diagramma di flusso non copre l'impatto delle policy basate sulle risorse e le negazioni implicite in altri tipi di policy.



*A session principal is either a role session or an IAM federated user session.

1. Rifiuta valutazione: per impostazione predefinita, tutte le richieste vengono rifiutate. Si tratta del cosiddetto rifiuto implicito. Il codice di AWS applicazione valuta tutte le politiche all'interno dell'account che si applicano alla richiesta. Questi includono AWS Organizations SCP, politiche basate sulle risorse, politiche basate sull'identità, limiti delle autorizzazioni IAM e politiche di sessione. In tutte le policy, il codice di attuazione cerca un'istruzione `Deny` applicabile alla richiesta. Questa azione si chiama rifiuto esplicito. Se il codice di attuazione trova anche un solo rifiuto esplicito applicabile, restituisce Deny (Rifiuta) come decisione finale. Se non c'è un rifiuto esplicito, la valutazione del codice di attuazione continua.
2. Organizations SCP: quindi il codice di applicazione valuta le politiche di controllo del AWS Organizations servizio (SCP) che si applicano alla richiesta. Le SCP si applicano alle entità dell'account in cui sono collegate le SCP. Se il codice di attuazione non trova istruzioni `Allow` applicabili nelle SCP, la richiesta viene rifiutata esplicitamente, anche se il rifiuto è implicito. Il codice di attuazione restituisce Deny (Rifiuta) come decisione finale. Se non c'è alcuna SCP oppure se l'SCP consente l'operazione richiesta, la valutazione del codice di attuazione continua.
3. Policy basate sulle risorse: all'interno dello stesso account, le policy basate sulle risorse influiscono sulla valutazione delle policy in modo diverso a seconda del tipo di principale che accede alla risorsa e al principale consentito nella policy basata sulle risorse. A seconda del tipo di principale, un `Allow` in una policy basata sulle risorse può comportare una decisione definitiva di `Allow`, anche se è presente un rifiuto implicito in una policy basata su identità, un limite delle autorizzazioni o una policy di sessione.

   Per la maggior parte delle risorse, è necessario solo un permesso esplicito per il principale in una policy basata sulle identità o una policy basata sulle risorse per concedere l'accesso. Le policy di affidabilità dei ruoli IAM e le policy delle chiavi KMS sono eccezioni a questa logica, perché devono consentire esplicitamente l'accesso per i principali.

   La policy basata sulle risorse differisce dagli altri tipi di policy se il principale specificato è un utente IAM, un ruolo IAM o un principale di sessione. I principi della sessione includono sessioni come ruolo IAM o una sessione come utente federato IAM. Se una policy basata sulle risorse concede l'autorizzazione direttamente all'utente IAM o al principale di sessione che sta effettuando la richiesta, un rifiuto implicito in una policy basata sull'identità, un limite di autorizzazioni o una policy di sessione non influiscono sulla decisione finale.

   La seguente tabella consente di comprendere l'impatto delle policy basate sulle risorse per diversi tipi di principali quando i rifiuti impliciti sono presenti nelle policy basate su identità, nei limiti delle autorizzazioni e nelle policy di sessione.

Policy basate sulle risorse e rifiuti impliciti in altri tipi di policy (stesso account)

Principale che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizzazioni	Policy di sessione	Risultato	Motivo
Ruolo IAM	Non applicabile	Non applicabile	Non applicabile	Non applicabile	Non applicabile	Un ruolo di per sé non può effettuare una richiesta. Le richieste vengono effettuate con la sessione come ruolo dopo l'assunzione di un ruolo.

Principal e che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizzazioni	Policy di sessione	Risultato	Motivo
Sessione come ruolo IAM	Consente il ruolo ARN	Rifiuto implicito	Rifiuto implicito	Rifiuto implicito	DENY	Il limite delle autorizzazioni e le policy di sessione vengono valutati come parte della decisione finale. Una negazione implicita in entrambe le policy si traduce in una decisione DENY.

Principal e che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizza zioni	Policy di sessione	Risultato	Motivo
Sessione come ruolo IAM	Consente l'ARN della sessione come ruolo	Rifiuto implicito	Rifiuto implicito	Rifiuto implicito	PERMETTI	Le autorizza zioni sono concesse direttame nte alla sessione. Altri tipi di policy non influisco no sulla decisione.
Utente IAM	Consente l'ARN dell'utente IAM	Rifiuto implicito	Rifiuto implicito	Non applicabile	PERMETTI	Le autorizza zioni vengono concesse direttame nte all'utente. Altri tipi di policy non influisco no sulla decisione.

Principal e che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizza zioni	Policy di sessione	Risultato	Motivo
Utente federato IAM (`GetFedera tionToken`)	Consente l'ARN dell'utente IAM	Rifiuto implicito	Rifiuto implicito	Rifiuto implicito	DENY	Un rifiuto implicito nel limite delle autorizza zioni o nella policy di sessione determina un RIFIUTA.
Utente federato IAM (`GetFedera tionToken`)	Consente l'ARN della sessione come utente federato IAM	Rifiuto implicito	Rifiuto implicito	Rifiuto implicito	PERMETTI	Le autorizza zioni sono concesse direttame nte alla sessione. Altri tipi di policy non influisco no sulla decisione.

Principal e che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizza zioni	Policy di sessione	Risultato	Motivo
utente root	Consente l'ARN dell'utente root	Non applicabile	Non applicabile	Non applicabile	PERMETTI	L'utente root dispone di accesso completo e illimitat o a tutte le risorse nell' Account AWS. Per informazi oni su come controllare l'accesso dell'uten te root per gli account in AWS Organizat ions, consultar e le Policy di controllo dei servizi (SCP) nella Guida per l'utente di

Principale che effettua la richiesta	Policy basata su risorse	Policy basata su identità	Limite delle autorizzazioni	Policy di sessione	Risultato	Motivo
						Organizations.
AWS preside del servizio	Consente un principale AWS di servizio	Non applicabile	Non applicabile	Non applicabile	PERMETTI	Quando una policy basata sulle risorse concede le autorizzazioni direttamente a un principale di servizio AWS, altri tipi di policy non influiscono sulla decisione.

- Ruolo IAM: i criteri basati sulle risorse che concedono le autorizzazioni a un ARN del ruolo IAM sono limitati da un rifiuto implicito in un limite delle autorizzazioni o in una policy di sessione.

  Esempio di ARN di ruolo

  ```
  arn:aws:iam::111122223333:role/examplerole
  ```

- Sessione come ruolo IAM: all'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN della sessione come ruolo IAM concedono le autorizzazioni direttamente alla sessione come ruolo assunto. Le autorizzazioni concesse direttamente a una

sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione. Quando si assume un ruolo e si effettua una richiesta, il principale che effettua la richiesta è l'ARN della sessione come ruolo IAM e non l'ARN del ruolo stesso.

Esempio di ARN della sessione come ruolo

```
arn:aws:sts::111122223333:assumed-role/examplerole/examplerolesessionname
```

- Utente IAM: all'interno dello stesso account, le politiche basate sulle risorse che concedono autorizzazioni all'ARN di un utente IAM (ovvero, non una sessione come utente federato) non sono limitate da un rifiuto implicito in una policy basata su identità o in un limite delle autorizzazioni.

  Esempio di ARN dell'utente IAM

  ```
  arn:aws:iam::111122223333:user/exampleuser
  ```

- Sessioni come utente federato IAM: una sessione come utente federato IAM è una sessione creata chiamando `GetFederationToken`. Quando un utente federato effettua una richiesta, il principale che effettua la richiesta è l'ARN dell'utente federato e non l'ARN dell'utente IAM che ha eseguito la federazione. All'interno dello stesso account, le policy basate sulle risorse che concedono le autorizzazioni all'ARN dell'utente federato concedono le autorizzazioni direttamente alla sessione. Le autorizzazioni concesse direttamente a una sessione non sono limitate da un rifiuto implicito in una policy basata su identità, da un limite delle autorizzazioni o da una policy di sessione.

  Tuttavia, se una policy basata sulle risorse concede l'autorizzazione all'ARN dell'utente IAM che ha eseguito la federazione, le richieste fatte dall'utente federato durante la sessione sono limitate da un rifiuto implicito in un limite di autorizzazione o in una policy di sessione.

  Esempio di ARN della sessione come utente federato IAM

  ```
  arn:aws:sts::111122223333:federated-user/exampleuser
  ```

4. Policy basate su identità: il codice verifica quindi le policy basate su identità per il principale. Per un utente IAM, queste includono le policy utente e le policy dei gruppi a cui appartiene l'utente. Se non ci sono policy basate su identità o istruzioni nelle policy basata su identità che consentono l'operazione richiesta, la richiesta viene rifiutata implicitamente e il codice restituisce Deny (Rifiuta)

come decisione finale. Se un'istruzione in qualsiasi policy basata su identità applicabile consente l'operazione richiesta, il codice continua.

5. Limiti delle autorizzazioni IAM: il codice controlla quindi se l'entità IAM utilizzata dall'entità ha un limite di autorizzazioni. Se la policy utilizzata per impostare il limite delle autorizzazioni non consente l'operazione richiesta, la richiesta viene rifiutata implicitamente. Il codice restituisce Deny (Rifiuta) come decisione finale. Se non c'è alcun limite delle autorizzazioni oppure se il limite delle autorizzazioni consente l'operazione richiesta, il codice continua.
6. Policy di sessione: il codice verifica quindi se il principale è un principale di sessione. I principali di sessione includono una sessione come ruolo IAM o una sessione come utente federato IAM. Se il principale non è un principale di sessione, il codice di attuazione restituisce Allow (Consenti) come decisione finale.

   Per i principali di sessione, il codice verifica se una policy di sessione è passata nella richiesta. Puoi passare una policy di sessione mentre usi l' AWS API AWS CLI or per ottenere credenziali temporanee per un ruolo o un utente federato IAM.

   - Se una policy di sessione è presente e non consente l'operazione richiesta, la richiesta viene rifiutata implicitamente. Il codice restituisce Deny (Rifiuta) come decisione finale.
   - Se non esiste una policy di sessione, il codice verifica se il principale è una sessione come ruolo. Se il principale è una sessione come ruolo, la richiesta è autorizzata. In caso contrario, la richiesta viene negata implicitamente e il codice restituisce Rifiuta come decisione finale.
   - Se una policy di sessione è presente e consente l'operazione richiesta, il codice di attuazione restituisce Consenti come decisione finale.
7. Errori: se il codice di AWS applicazione rileva un errore in qualsiasi momento durante la valutazione, genera un'eccezione e si chiude.

Esempio di valutazione delle policy basate su identità e delle policy basate su risorse

I tipi di policy più comuni sono quelle basate su identità e quelle basate su risorse. Quando viene richiesto l'accesso a una risorsa, AWS valuta tutte le autorizzazioni concesse dalle politiche per almeno un account Allow all'interno dello stesso account. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione.

> **Important**
>
> Se la policy basata sull'identità o la policy basata sulle risorse all'interno dello stesso account consente la richiesta e l'altra no, la richiesta è comunque consentita.

Supponiamo che Carlos, con il nome utente `carlossalazar`, voglia salvare un file nel bucket `carlossalazar-logs` di Amazon S3.

Supponi inoltre che la policy seguente sia collegata all'utente IAM `carlossalazar`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ListRead",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAccessPoints",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "AllowS3Self",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::carlossalazar/*",
                "arn:aws:s3:::carlossalazar"
            ]
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::*log*"
        }
    ]
}
```

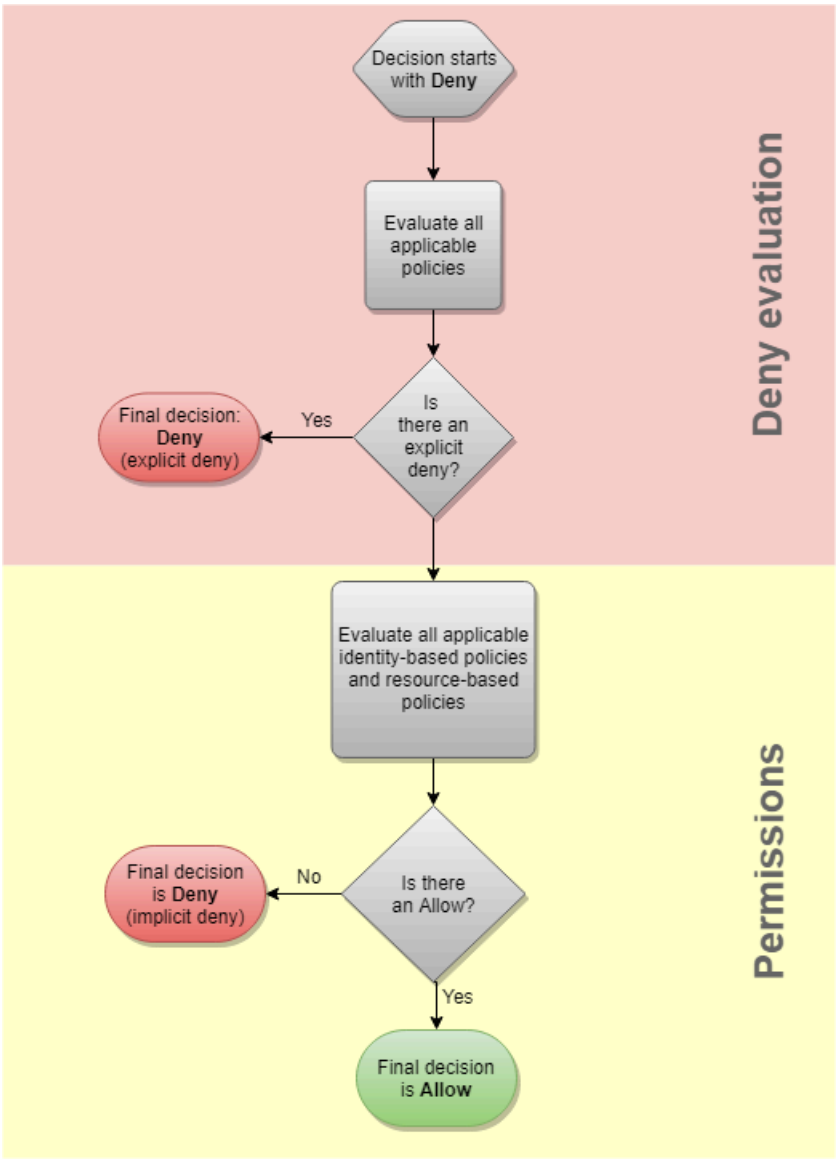
L'istruzione `AllowS3ListRead` in questa policy consente a Carlos di visualizzare un elenco di tutti i bucket nell'account. L'istruzione `AllowS3Self` consente a Carlos l'accesso completo al bucket con lo stesso nome usato per il nome utente. L'istruzione `DenyS3Logs` nega a Carlos l'accesso a qualsiasi bucket di S3 il cui nome includa `log`.

Inoltre, la seguente policy basata su risorse (detta policy del bucket) viene collegata al bucket `carlossalazar`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/carlossalazar"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::carlossalazar/*",
                "arn:aws:s3:::carlossalazar"
            ]
        }
    ]
}
```

Questa policy specifica che solo l'utente `carlossalazar` può accedere al bucket `carlossalazar`.

Quando Carlos richiede di salvare un file nel `carlossalazar-logs` bucket, AWS determina quali politiche si applicano alla richiesta. In questo caso, sono applicabili solo la policy basata su identità e la policy basata su risorse. Entrambe sono policy di autorizzazione. Poiché non ci sono limiti di autorizzazione applicabili, la logica di valutazione viene ridotta a quanto segue.



AWS verifica innanzitutto la presenza di un'`Deny`istruzione che si applichi al contesto della richiesta. Ne trova una, perché la policy basata su identità rifiuta esplicitamente a Carlos l'accesso a qualsiasi bucket di S3 utilizzato per la creazione di log. A Carlos viene negato l'accesso.

Supponiamo che poi si renda conto del suo errore e cerchi di salvare il file nel `carlossalazar` bucket. AWS verifica la presenza di una `Deny` dichiarazione e non la trova. Verifica quindi le policy di autorizzazione. Sia la policy basata sull'identità che la policy basata sulle risorse consentono la richiesta. Pertanto, AWS consente la richiesta. Se uno dei due avesse rifiutato esplicitamente l'istruzione, la richiesta sarebbe stata negata. Se uno dei tipi di policy consente la richiesta e l'altro no, la richiesta è comunque consentita.

Differenza tra rifiuto esplicito e implicito

Una richiesta genera un rifiuto esplicito se policy applicabile include un'istruzione `Deny`. Se le policy applicabili a una richiesta includono un'istruzione `Allow` e un'istruzione `Deny`, l'istruzione `Deny` prevale sull'istruzione `Allow`. La richiesta viene rifiutata esplicitamente.

Una rifiuto implicito si verifica quando non c'è un'istruzione `Deny` applicabile ma non c'è neanche un'istruzione `Allow` applicabile. Poiché a un principale IAM viene rifiutato l'accesso per impostazione predefinita, questo deve essere autorizzato esplicitamente a eseguire un'operazione. In caso contrario, l'accesso viene negato implicitamente.

Quando progetti una strategia di autorizzazione, devi creare policy con istruzioni `Allow` per consentire alle entità principali di eseguire richieste. Tuttavia, puoi scegliere qualsiasi combinazione di rifiuti espliciti e impliciti.

Ad esempio, è possibile creare la seguente policy che include operazioni consentite, operazioni rifiutate implicitamente e operazioni rifiutate esplicitamente. La dichiarazione `AllowGetList` permette l'accesso in sola lettura alle operazioni IAM che iniziano con i prefissi `Get` e `List`. Tutte le altre azioni in IAM, come `iam:CreatePolicy`, sono rifiutate implicitamente. La dichiarazione `DenyReports` impedisce esplicitamente l'accesso ai report IAM impedendo l'accesso alle operazioni che includono il suffisso `Report`, come `iam:GetOrganizationsAccessReport`. Se qualcuno aggiunge un'altra policy a questo principale per concedere l'accesso ai report IAM, come `iam:GenerateCredentialReport`, le richieste relative ai report vengono ancora rifiutate a causa di questo rifiuto esplicito.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowGetList",
            "Effect": "Allow",
            "Action": [
                "iam:Get*",
                "iam:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenyReports",
            "Effect": "Deny",
            "Action": "iam:*Report",
```

```
            "Resource": "*"
        }
    ]
}
```

Logica di valutazione della policy multiaccount

Puoi consentire a un principale in un account di accedere alle risorse in un secondo account. Questo è chiamato accesso tra account. Quando consenti l'accesso tra account, l'account in cui si trova il principale viene denominato l'account attendibile . L'account in cui si trova la risorsa è l'account che concede fiducia .

Per consentire l'accesso tra account, collega una policy basata sulle risorse alla risorsa che desideri condividere. Devi inoltre collegare una policy basata sull'identità all'identità che agisce come il principale nella richiesta. La policy basata su risorse nell'account che concede fiducia deve specificare il principale dell'account attendibile che avrà accesso alla risorsa. Puoi specificare l'intero account o i relativi utenti IAM, gli utenti federati, i ruoli IAM o le sessioni del ruolo assunto. È inoltre possibile specificare un AWS servizio come principale. Per ulteriori informazioni, consulta Specifica di un'entità principale.

La policy basata su identità del principale deve consentire l'accesso richiesto alla risorsa nel servizio che concede fiducia. A questo scopo, specifica l'ARN della risorsa o consenti l'accesso a tutte le risorse (*).

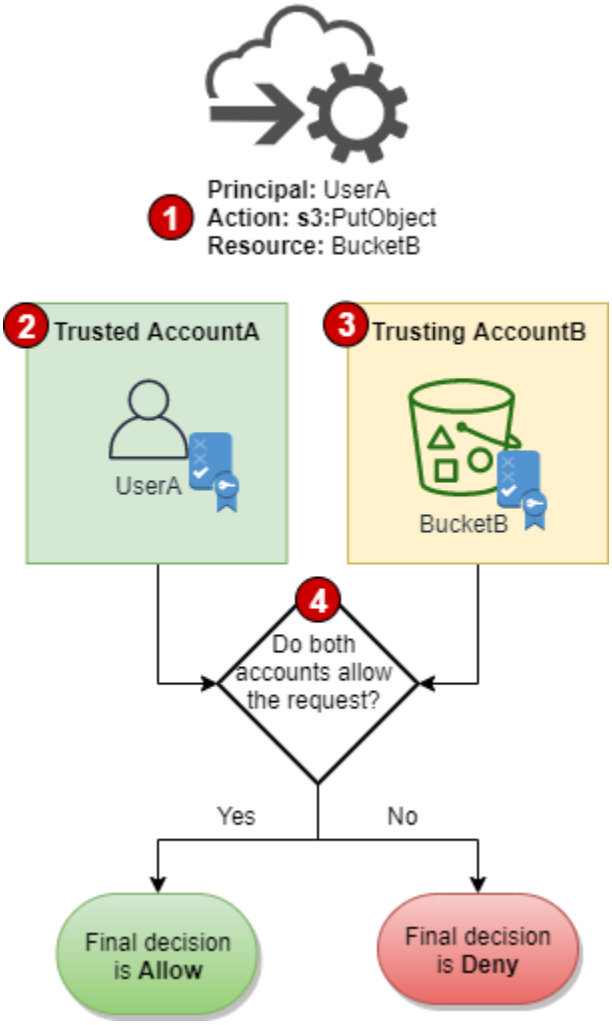
In IAM, puoi collegare una policy basata sulle risorse a un ruolo IAM per consentire ai principali in altri account di assumere tale ruolo. La policy basata sulle risorse del ruolo è denominata policy di attendibilità del ruolo. Dopo aver assunto tale ruolo, i principali consentiti possono utilizzare le credenziali temporanee risultanti per accedere a più risorse nell'account. Questo accesso è definito nella policy di autorizzazioni basata su identità del ruolo. Per informazioni sul perché consentire l'accesso tra account utilizzando i ruoli è diverso dal consentire l'accesso tra account utilizzando altre policy basate sulle risorse, consulta Accesso alle risorse multi-account in IAM.

> **Important**
>
> Altri servizi possono influire sulla logica di valutazione dei criteri. Ad esempio, AWS Organizations supporta le politiche di controllo dei servizi che possono essere applicate ai responsabili di uno o più account. AWS Resource Access Manager supporta frammenti di policy che controllano le azioni che i mandanti sono autorizzati a eseguire sulle risorse condivise con loro.

Determinare se una richiesta tra account è consentita

Per le richieste tra account, il richiedente nell'`AccountA` attendibile deve disporre di una policy basata su identità. Tale policy deve consentire di effettuare una richiesta alla risorsa nell' che concede fiducia `AccountB`. Inoltre, la policy basata sulle risorse nell'`AccountB` deve consentire al richiedente nell'`AccountA` di accedere alla risorsa.

Quando effettui una richiesta tra più account, AWS esegue due valutazioni. AWS valuta la richiesta nell'account fiduciario e nell'account fidato. Per ulteriori informazioni su come una richiesta viene valutata all'interno di un singolo account, consulta Determinazione se una richiesta è consentita o rifiutata in un account. La richiesta è consentita solo se entrambe le valutazioni restituiscono come decisione `Allow`.



1. Quando un principale in un account effettua una richiesta per accedere a una risorsa in un altro account, questa è una richiesta tra account.

2. Il principale che esegue la richiesta esiste nell'account attendibile (`AccountA`). Quando AWS valuta questo account, controlla la policy basata su identità e le eventuali policy che possono limitare una policy basata su identità. Per ulteriori informazioni, consulta Valutazione delle policy in un singolo account.
3. La risorsa richiesta esiste nell'account che concede fiducia (`AccountB`). Quando AWS valuta questo account, controlla la policy basata sulle risorse collegata alla risorsa richiesta e le eventuali policy che possono limitare una policy basata sulle risorse. Per ulteriori informazioni, consulta Valutazione delle policy in un singolo account.
4. AWS consente la richiesta solo se entrambe le valutazioni delle politiche dell'account consentono la richiesta.

Esempio di valutazione della policy multiaccount

Nell'esempio seguente viene illustrato uno scenario in cui a un utente in un account vengono concesse autorizzazioni da una policy basata sulle risorse in un secondo account.

Supponiamo che Carlos sia uno sviluppatore con un utente IAM denominato `carlossalazar` nell'account 111111111111. Vuole salvare un file nel bucket `Production-logs` di Amazon S3 nell'account 222222222222.

Supponi inoltre che la policy seguente sia collegata all'utente IAM `carlossalazar`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ListRead",
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Sid": "AllowS3ProductionObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object*",
            "Resource": "arn:aws:s3:::Production/*"
        },
        {
            "Sid": "DenyS3Logs",
            "Effect": "Deny",
```

```
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::*log*",
                "arn:aws:s3:::*log*/*"
            ]
        }
    ]
}
```

L'istruzione `AllowS3ListRead` in questa policy consente a Carlos di visualizzare un elenco di tutti i bucket in Amazon S3. L'istruzione `AllowS3ProductionObjectActions` consente a Carlos l'accesso completo al bucket `Production`. L'istruzione `DenyS3Logs` nega a Carlos l'accesso a qualsiasi bucket di S3 il cui nome includa `log`. Nega anche l'accesso a tutti gli oggetti in quei bucket.

Inoltre, la seguente policy basata sulle risorse (denominata policy del bucket) è collegata al bucket `Production` nell'account 222222222222.

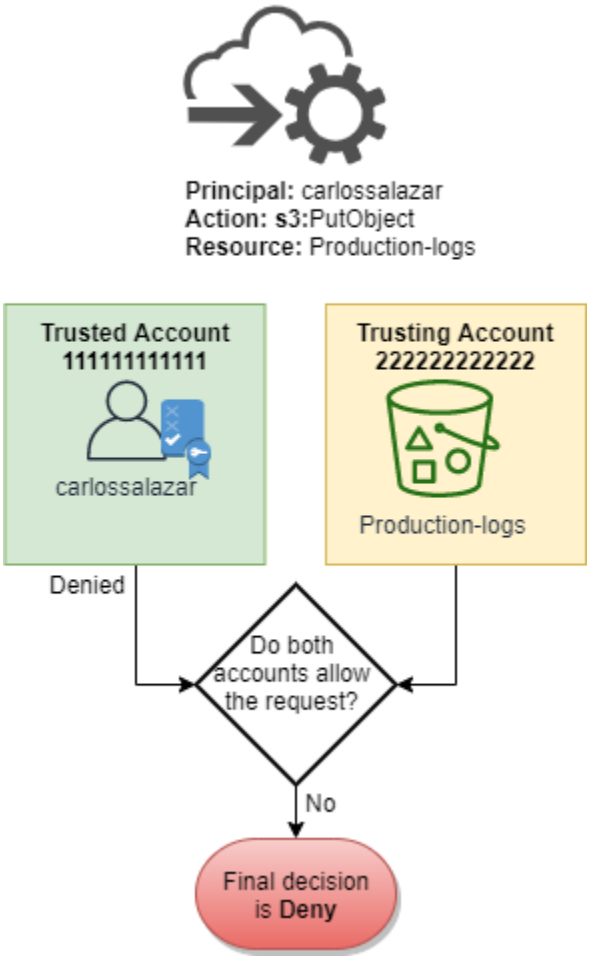
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:PutObject*",
                "s3:ReplicateObject",
                "s3:RestoreObject"
            ],
            "Principal": { "AWS": "arn:aws:iam::111111111111:user/carlossalazar" },
            "Resource": "arn:aws:s3:::Production/*"
        }
    ]
}
```

Questo criterio consente all'utente `carlossalazar` di accedere agli oggetti nel bucket `Production`. Può creare e modificare, ma non eliminare gli oggetti nel bucket. Non riesce a gestire il bucket da solo.

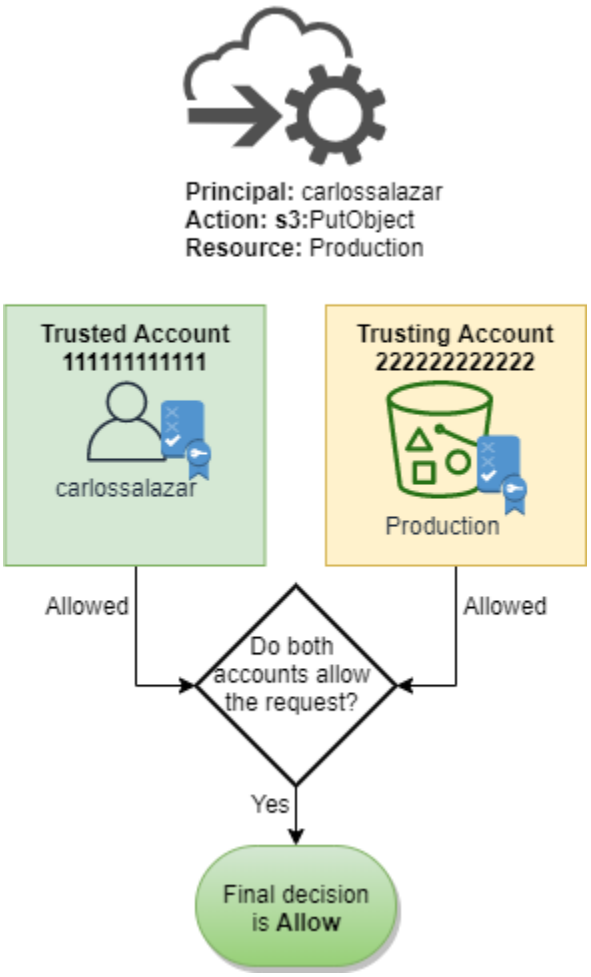
Quando Carlos esegue la richiesta di salvataggio di un file nel bucket `Production-logs`, AWS determina quali siano le policy applicabili alla richiesta. In questo caso, la policy basata su identità collegata all'utente `carlossalazar` è la sola policy valida nell'account `111111111111`.

Nell'account 222222222222, non esiste una policy basata sulle risorse collegata al bucket `Production-logs`. Quando AWS valuta l'account 111111111111, restituisce come decisione `Deny`. Questo perché l'istruzione `DenyS3Logs` nella policy basata su identità nega esplicitamente l'accesso a qualsiasi bucket di log. Per ulteriori informazioni su come una richiesta viene valutata all'interno di un singolo account, consulta Determinazione se una richiesta è consentita o rifiutata in un account.

Poiché la richiesta viene negata esplicitamente all'interno di uno degli account, la decisione finale è di negare la richiesta.



Supponiamo che Carlos si renda conto del suo errore e cerchi di salvare il file nel `Production` bucket. AWS controlla innanzitutto l'account 111111111111 per determinare se la richiesta è consentita. Si applica solo la politica basata sull'identità e consente la richiesta. AWS quindi controlla l'account. 222222222222 Vale solo la policy basata sulle risorse collegata al bucket `Production` e consente la richiesta. Poiché entrambi gli account consentono la richiesta, la decisione finale è di consentire la richiesta.



Sintassi del linguaggio della policy JSON IAM

Questa pagina riporta una sintassi formale per il linguaggio utilizzato per creare le policy JSON in IAM. Presentiamo questa grammatica in modo che sia possibile comprendere come costruire e convalidare le policy.

Per esempi di policy, consultare i seguenti argomenti:

- Policy e autorizzazioni in IAM
- Esempi di policy basate su identità IAM
- Policy di esempio per lavorare nella console Amazon EC2 ed esempi di policy per lavorare con la CLI, l'Amazon EC2 AWS CLI o un SDK AWS nella Amazon EC2 User Guide.
- Esempi di policy del bucket e Esempi di policy utente nella Guida per l'utente di Amazon Simple Storage Service.

Per esempi di politiche utilizzate in altri AWS servizi, consulta la documentazione relativa a tali servizi.

Argomenti

Il linguaggio di policy e JSON

Le policy sono espresse in JSON. Quando crei o modifichi una policy JSON, IAM può eseguire la convalida delle policy per facilitare la creazione di una policy efficace. IAM identificherà gli errori di sintassi JSON, mentre IAM Access Analyzer fornisce ulteriori controlli delle policy con suggerimenti che consentono di perfezionare ulteriormente le policy. Per ulteriori informazioni sulla convalida delle policy, consulta Convalida delle policy IAM. Per ulteriori informazioni cui controlli delle policy di IAM Access Analyzer e sui suggerimenti utili, consulta Convalida delle policy di IAM Access Analyzer.

In questo documento, non forniamo una descrizione completa di ciò che costituisce un JSON valido. Tuttavia, alcune regole JSON di base:

- È consentito spazio vuoto tra singole entità.
- I valori sono racchiusi tra virgolette. Le virgolette sono facoltative per valori numerici e booleani.
- Molti elementi (ad esempio `action_string_list` e `resource_string_list`) possono richiedere un array JSON come valore. Gli array possono richiedere uno o più valori. Se più di un valore è incluso, l'array è tra parentesi quadre (`[` e `]`) e delimitato da virgole, come nell'esempio seguente:

  ```
  "Action" : ["ec2:Describe*","ec2:List*"]
  ```
- I tipi di dati JSON di base (booleano, numero e stringa) sono definiti in RFC 7159.

Convenzioni utilizzate in questa sintassi

Le convenzioni seguenti vengono utilizzate in questa grammatica:

- I seguenti caratteri sono token JSON e sono inclusi nelle policy:

```
{ } [ ] " , :
```

- I seguenti caratteri sono caratteri speciali nella grammatica e non sono inclusi nelle policy:

  ```
  = < > ( ) |
  ```

- Se un elemento permette più valori, è indicato utilizzando valori ripetuti, un delimitatore di virgole e puntini di sospensione (`...`). Esempi:

  ```
  [<action_string>, <action_string>, ...]
  ```

  ```
  <principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }
  ```

  Se più valori sono consentiti, è anche valido per includere un solo valore. Per un solo valore, la virgola finale deve essere omessa. Se l'elemento richiede un array (contrassegnato con [e]), ma solo un valore è incluso, le parentesi sono facoltative. Esempi:

  ```
  "Action": [<action_string>]
  ```

  ```
  "Action": <action_string>
  ```

- Un punto di domanda (?) in seguito a un elemento indica che l'elemento è facoltativo. Esempio:

  ```
  <version_block?>
  ```

  Tuttavia, assicurarsi di fare riferimento alle note che seguono l'inserzione sulla grammatica sugli elementi opzionali.

- Una linea verticale (|) tra elementi indica alternative. Nella grammatica, le parentesi definiscono la portata delle alternative. Esempio:

  ```
  ("Principal" | "NotPrincipal")
  ```

- Gli elementi che devono essere stringhe letterali vengono racchiusi tra virgolette ("). Esempio:

  ```
  <version_block> = "Version" : ("2008-10-17" | "2012-10-17")
  ```

Per ulteriori note, consultare Note sulla sintassi delle policy in seguito all'inserzione sulla grammatica.

Grammatica

La seguente inserzione descrive il linguaggio grammaticale della policy. Per le convenzioni utilizzate nell'inserzione, consultare la sezione precedente. Per ulteriori informazioni, consultare le seguenti note:

> **Note**
>
> Questa grammatica descrive le policy contrassegnate con una versione di `2008-10-17` e `2012-10-17`. Un elemento di policy `Version` è diverso da una versione di policy. L'elemento di policy `Version` viene utilizzato all'interno di una policy e definisce la versione del linguaggio di policy. Diversamente, una versione della policy viene creata quando si apportano modifiche alla policy gestita dal cliente in IAM. La policy modificata non viene sovrascritta a quella precedente. IAM crea invece una nuova versione della policy gestita. Per ulteriori informazioni sull'elemento di policy `Version`, consultare Elementi delle policy JSON IAM: Version. Per ulteriori informazioni sulle versioni di policy, consultare the section called “Controllo delle versioni delle policy IAM”.

```
policy  = {
     <version_block?>
     <id_block?>
     <statement_block>
}

<version_block> = "Version" : ("2008-10-17" | "2012-10-17")

<id_block> = "Id" : <policy_id_string>

<statement_block> = "Statement" : [ <statement>, <statement>, ... ]

<statement> = {
    <sid_block?>,
    <principal_block?>,
    <effect_block>,
    <action_block>,
    <resource_block>,
    <condition_block?>
}

<sid_block> = "Sid" : <sid_string>
```

```
<effect_block> = "Effect" : ("Allow" | "Deny")

<principal_block> = ("Principal" | "NotPrincipal") : ("*" | <principal_map>)

<principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }

<principal_map_entry> = ("AWS" | "Federated" | "Service" | "CanonicalUser") :
    [<principal_id_string>, <principal_id_string>, ...]

<action_block> = ("Action" | "NotAction") :
    ("*" | [<action_string>, <action_string>, ...])

<resource_block> = ("Resource" | "NotResource") :
    : ("*" | <resource_string> | [<resource_string>, <resource_string>, ...])

<condition_block> = "Condition" : { <condition_map> }
<condition_map> = {
  <condition_type_string> : { <condition_key_string> : <condition_value_list> },
  <condition_type_string> : { <condition_key_string> : <condition_value_list> }, ...
}
<condition_value_list> = [<condition_value>, <condition_value>, ...]
<condition_value> = (<condition_value_string> | <condition_value_string> |
 <condition_value_string>)
```

Note sulla sintassi delle policy

- Una singola policy può contenere una gamma di istruzioni.
- Le policy hanno una dimensione massima tra 2048 e 10.240 caratteri, in base a quale entità la policy è collegata. Per ulteriori informazioni, consulta IAM e AWS STS quote. I calcoli delle dimensioni della policy non includono spazi vuoti.
- I singoli elementi non devono contenere più istanze della stessa chiave. Ad esempio, non è possibile includere il blocco `Effect` due volte nella stessa istruzione.
- I blocchi possono essere visualizzati in qualsiasi ordine. Ad esempio, `version_block` può seguire `id_block` in una policy. Analogamente, `effect_block`, `principal_block`, `action_block` può comparire in qualsiasi ordine all'interno di un'istruzione.
- `id_block` è facoltativo nelle policy basate su risorse. Non deve essere incluso nelle policy basate sulle identità.

- L'elemento `principal_block` è obbligatorio nelle policy basate su risorse (ad esempio, nelle policy del bucket Amazon S3) e nelle policy di attendibilità per i ruoli IAM. Non deve essere incluso nelle policy basate sulle identità.
- L'elemento `principal_map` nelle policy di bucket Amazon S3 può includere l'ID `CanonicalUser`. La maggior parte delle policy basate su risorse non supporta questa mappatura. Per ulteriori informazioni sull'utilizzo dell'ID utente canonico in una policy di bucket, consulta Specifica di un principale in una policy nella Guida per l'utente di Amazon Simple Storage Service.
- Ogni valore di stringa (`policy_id_string`, `sid_string`, `principal_id_string`, `action_string`, `resource_string`, `condition_type_string`, `condition_key_string` e la versione di stringa di `condition_value`) può avere propri valori consentiti specifici, restrizioni di lunghezza minima e massima e un formato interno necessario.

Note sui valori di stringa

Questa sezione fornisce ulteriori informazioni sui valori di stringa utilizzati in diversi elementi in una policy.

`action_string`

Consiste in uno spazio dei nomi di un servizio, due punti e il nome di un'azione. I nomi delle operazioni possono includere caratteri jolly. Esempi:

```
"Action":"ec2:StartInstances"

"Action":[
  "ec2:StartInstances",
  "ec2:StopInstances"
]

"Action":"cloudformation:*"

"Action":"*"

"Action":[
  "s3:Get*",
  "s3:List*"
]
```

`policy_id_string`

Fornisce un modo per includere informazioni sulla policy complessiva. Alcuni servizi, ad esempio Amazon SQS e Amazon SNS, utilizzano l'elemento `Id` in modi riservati. Salvo diversamente limitato da un singolo servizio, policy_id_string può includere spazi. Alcuni servizi richiedono che questo valore sia univoco in un account AWS .

> **Note**
>
> L'endpoint `id_block` è consentito nelle policy basate su risorse, ma non nelle policy basate sulle identità.

Non esiste alcun limite alla lunghezza, anche se questa stringa contribuisce alla lunghezza complessiva della policy, che è limitata.

```
"Id":"Admin_Policy"

"Id":"cd3ad3d9-2776-4ef1-a904-4c229d1642ee"
```

`sid_string`

Fornisce un modo per includere informazioni su una istruzione individuale. Per le policy IAM, i caratteri alfanumerici di base (A-Z, a-z, 0-9) sono i soli valori consentiti nel valore `Sid`. Altri servizi AWS che supportano policy basate su risorse possono avere altri requisiti per il valore `Sid`. Ad esempio, alcuni servizi richiedono che questo valore sia univoco all'interno di un Account AWS valore e alcuni servizi consentono caratteri aggiuntivi come gli spazi nel `Sid` valore.

```
"Sid":"1"

"Sid": "ThisStatementProvidesPermissionsForConsoleAccess"
```

`principal_id_string`

Fornisce un modo per specificare un principale utilizzando l'Amazon Resource Name (ARN) dell'utente IAM Account AWS, del ruolo IAM, dell'utente federato o dell'utente con ruolo assunto. In caso Account AWS affermativo, puoi anche utilizzare il modulo breve `AWS:`*`accountnumber`* anziché l'ARN completo. Per tutte le opzioni, tra cui ruoli assunti, servizi AWS e così via, consulta Specifica di un'entità principale.

È possibile utilizzare * solo per specificare "tutti/anonimi". Non è possibile utilizzarlo per specificare una parte del nome o di ARN.

`resource_string`

Nella maggior parte dei casi, è composto da un Amazon Resource Name (ARN).

```
"Resource":"arn:aws:iam::123456789012:user/Bob"

"Resource":"arn:aws:s3:::examplebucket/*"
```

`condition_type_string`

Identifica il tipo di condizione da testare, ad esempio `StringEquals`, `StringLike`, `NumericLessThan`, `DateGreaterThanEquals`, `Bool`, `BinaryEquals`, `IpAddress`, `ArnEquals` ecc. Per l'elenco completo dei tipi di condizione, consultare Elementi della policy JSON IAM: operatori di condizione.

```
"Condition": {
  "NumericLessThanEquals": {
    "s3:max-keys": "10"
  }
}

"Condition": {
  "Bool": {
    "aws:SecureTransport": "true"
  }
}

"Condition": {
  "StringEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
   }
}
```

`condition_key_string`

Identifica la chiave di condizione il cui valore verrà testato per determinare se la condizione è soddisfatta. AWS definisce un set di chiavi di condizione disponibili in tutti i AWS servizi, tra cui `aws:PrincipalTypeaws:SecureTransport`, e`aws:userid`.

Per un elenco delle chiavi di AWS condizione, vedereAWS chiavi di contesto della condizione globale. Per le chiavi di condizione specifiche per un servizio, consultare la documentazione per quel servizio, tra cui quanto segue:

- Specifica delle condizioni in una policy nella Guida per l'utente di Amazon Simple Storage Service
- Policy IAM per Amazon EC2 nella Guida per l'utente di Amazon EC2.

```
"Condition":{
  "Bool": {
      "aws:SecureTransport": "true"
   }
}

"Condition": {
  "StringNotEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
   }
}

"Condition": {
  "StringEquals": {
    "aws:ResourceTag/purpose": "test"
  }
}
```

`condition_value_string`

Identifica il valore di condition_key_string che determina se la condizione è soddisfatta. Per un elenco completo di valori validi per un tipo di condizione, consulta Elementi della policy JSON IAM: operatori di condizione.

```
"Condition":{
  "ForAnyValue:StringEquals": {
  "dynamodb:Attributes": [
   "ID",
   "PostDateTime"
         ]
  }
}
```

AWS politiche gestite per le funzioni lavorative

Consigliamo di utilizzare le policy che concedono il privilegio minimo o che concedono solo le autorizzazioni richieste per eseguire un processo. Il modo più sicuro per concedere il privilegio minimo consiste nello scrivere una policy personalizzata con solo le autorizzazioni necessarie al team. È necessario creare un processo per consentire al team di richiedere ulteriori autorizzazioni quando necessario. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza.

Per iniziare ad aggiungere autorizzazioni alle tue identità IAM (utenti, gruppi di utenti e ruoli), puoi utilizzare. AWS politiche gestite AWS le politiche gestite coprono casi d'uso comuni e sono disponibili nel tuo. Account AWS AWS le politiche gestite non concedono i permessi con il privilegio minimo. Considera il rischio per la sicurezza di concedere ai principali più autorizzazioni di quelle necessarie per svolgere il proprio lavoro.

Puoi allegare policy AWS gestite, incluse le funzioni lavorative, a qualsiasi identità IAM. Per passare alle autorizzazioni con privilegi minimi, puoi eseguire AWS Identity and Access Management Access Analyzer per monitorare i principali con policy gestite. AWS Dopo aver appreso quali autorizzazioni stanno utilizzando, puoi scrivere una policy personalizzata o generare una policy con solo le autorizzazioni richieste per il team. È meno sicuro, ma offre maggiore flessibilità man mano che impari a utilizzare il tuo team. AWS

AWS le politiche gestite per le funzioni lavorative sono progettate per allinearsi strettamente alle funzioni lavorative comuni nel settore IT. È possibile utilizzare queste policy per concedere le autorizzazioni necessarie per eseguire le attività che ci si aspetta da qualcuno in una determinata funzione lavorativa. Queste policy consolidano le autorizzazioni per molti servizi in un'unica policy con la quale è più semplice collaborare rispetto ad avere le autorizzazioni disperse in molte policy.

Utilizzare i ruoli per combinare i servizi

Alcune policy utilizzano i ruoli di servizio IAM per aiutarti a sfruttare le funzionalità disponibili in altri AWS servizi. Queste politiche concedono l'accesso `aiam:passrole`, il che consente a un utente con la policy di trasferire un ruolo a un AWS servizio. Questo ruolo delega le autorizzazioni IAM al AWS servizio per eseguire azioni per tuo conto.

È necessario creare ruoli in base alle proprie esigenze. Ad esempio, la policy Network Administrator consente a un utente con la policy di passare un ruolo denominato «flow-logs-vpc» al servizio Amazon. CloudWatch CloudWatch utilizza quel ruolo per registrare e acquisire il traffico IP per i VPC creati dall'utente.

Per seguire le best practice di sicurezza, le policy per le funzioni lavorative includono filtri che limitano i nomi dei ruoli validi che possono essere passati. In questo modo è possibile evitare di concedere autorizzazioni non necessarie. Se gli utenti richiedono i ruoli di servizio opzionali, è necessario creare un ruolo che segue la convenzione di denominazione specificata nella policy. È possibile concedere le autorizzazioni al ruolo. Una volta completata questa operazione, l'utente può configurare il servizio per utilizzare il ruolo, concedendogli tutte le autorizzazioni che il ruolo fornisce.

Nelle seguenti sezioni, ogni nome di policy è un collegamento alla pagina dei dettagli della policy nella AWS Management Console. Qui è possibile visualizzare il documento della policy e riconsultare le autorizzazioni che concede.

Funzione processo dell'amministratore

AWS nome della politica gestita: AdministratorAccess

Caso d'uso: questo utente ha accesso completo e può delegare le autorizzazioni per ogni servizio e risorsa in AWS.

Aggiornamenti della politica: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede tutte le azioni per tutti i AWS servizi e per tutte le risorse dell'account. Per ulteriori informazioni sulla policy gestita, consulta AdministratorAccessla AWS Managed Policy Reference Guide.

> **Note**
>
> Prima che un utente o un ruolo IAM possa accedere alla AWS Billing and Cost Management console con le autorizzazioni previste da questa policy, devi prima attivare l'accesso a utenti e ruoli IAM. A tale scopo, seguire le istruzioni nella fase 1 del tutorial sulla delega dell'accesso alla console di fatturazione.

Funzione di processo di fatturazione

AWS nome della politica gestita: Billing

Caso d'uso: questo utente deve visualizzare i dati di fatturazione, impostare i pagamenti e autorizzarli. L'utente può monitorare i costi accumulati per l'intero AWS servizio.

Aggiornamenti della politica: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione policy: questa policy concede le autorizzazioni complete per gestire fatturazione, costi, metodi di pagamento, budget e report. Per ulteriori esempi di politiche di gestione dei costi, consulta gli esempi di AWS Billing politiche nella Guida AWS Billing and Cost Management per l'utente. Per ulteriori informazioni sulla politica gestita, consulta la Guida di riferimento alla fatturazione nella AWS Managed Policy.

> **Note**
>
> Prima che un utente o un ruolo IAM possa accedere alla AWS Billing and Cost Management console con le autorizzazioni previste da questa policy, devi prima attivare l'accesso a utenti e ruoli IAM. A tale scopo, seguire le istruzioni nella fase 1 del tutorial sulla delega dell'accesso alla console di fatturazione.

Funzione di processo dell'amministratore di database

AWS nome della policy gestita: DatabaseAdministrator

Caso d'uso: questo utente configura, configura e gestisce i database nel AWS cloud.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione policy: questa policy concede le autorizzazioni per creare, configurare e gestire i database. Include l'accesso a servizi di AWS database, come Amazon DynamoDB, Amazon Relational Database Service (RDS) e Amazon Redshift. Visualizza la policy per l'elenco completo di servizi di database supportati dalla policy. Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference DatabaseAdministratorGuide AWS .

Questa politica sulle funzioni lavorative supporta la possibilità di trasferire ruoli ai AWS servizi. La policy consente l'operazione `iam:PassRole` solo per i ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Ruoli di servizio IAM facoltativi per la funzione lavorativa di amministratore del database

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	Seleziona questa politica AWS gestita
Consentire all'utente di monitorare i database RDS	rds-monitoring-role	Ruolo Amazon RDS per il monitoraggio avanzato	Ruolo di Amazon EnhancedMonitoring RDS
Consenti di AWS Lambda monitorare il tuo database e accedere a database esterni	rdbms-lambda-access	Amazon EC2	AWSLambda_FullAccess
Consentire a Lambda di caricare file su Amazon S3 e su cluster Amazon Redshift con DynamoDB	lambda_exec_role	AWS Lambda	Creare una nuova policy gestita secondo quanto definito in AWS Big Data Blog
Consentire alle funzioni Lambda di agire come trigger per le tabelle DynamoDB	lambda-dynamodb-*	AWS Lambda	AWSLambda DynamoDBExecutionRole
Consentire alle funzioni Lambda di accedere ad Amazon RDS in un VPC	lambda-vpc-execution-role	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per	AWSLambda VPCAccessExecution Role

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	Seleziona questa politica AWS gestita
		gli sviluppatori di AWS Lambda	
Consenti AWS Data Pipeline l'accesso alle tue AWS risorse	DataPipelineDefaultRole	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	La AWS Data Pipeline documentazione elenca le autorizzazioni richieste per questo caso d'uso. Vedi i ruoli IAM per AWS Data Pipeline
Consentire alle applicazioni in esecuzione su istanze Amazon EC2 di accedere alle risorse AWS	DataPipelineDefaultResourceRuolo	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	Amazon EC2 RoleforData PipelineRole

Funzione di processo per data scientist

AWS nome della politica gestita: DataScientist

Caso d'uso: questo utente esegue attività e query Hadoop. L'utente, inoltre accede e analizza le informazioni per l'analisi dei dati e di business intelligence.

Aggiornamenti della politica: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare, gestire ed eseguire query su un cluster Amazon EMR ed eseguire analisi dei dati con strumenti come Amazon. QuickSight La policy include l'accesso a servizi di data scientist aggiuntivi, come Amazon EC2 AWS Data Pipeline, Amazon Kinesis, Amazon Machine Learning e. SageMaker Visualizza la policy per l'elenco completo dei servizi scientifici dei dati supportati dalla policy. Per ulteriori informazioni sulla policy gestita, consulta DataScientistla Managed Policy Reference Guide.AWS

Questa politica sulle funzioni lavorative supporta la possibilità di trasferire ruoli ai AWS servizi. Un'istruzione consente di passare qualsiasi ruolo a SageMaker. Un'altra istruzione consente l'operazione `iam:PassRole` solo per i ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Ruoli di servizio IAM facoltativi per la funzione lavorativa di tecnico addetto alla gestione dei dati

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consentire alle istanze Amazon EC2 l'accesso ai servizi e alle risorse idonei per i cluster	EMR-EC2_ DefaultRole	Amazon EMR per EC2	AmazonEla sticMapRe duceforRuolo EC2
Consentire ad Amazon EMR di accedere al servizio e alle risorse Amazon EC2 per i cluster	EMR_ DefaultRole	Amazon EMR	Amazon EMR _v2 ServicePolicy
Consenti a Kinesis Managed Service per Apache Flink di accedere alle origini dati in streaming	kinesis-*	Creare un ruolo con una policy di affidabilità secondo quanto definito in AWS Big Data Blog.	Consultare AWS Big Data Blog, che delinea quattro possibili opzioni, a seconda del caso d'uso
Consenti l'accesso alle tue risorse AWS Data Pipeline AWS	DataPipelineDefaul tRole	Creazione di un ruolo con una policy di attendibi	La AWS Data Pipeline documenta zione elenca le

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da selezionare
		lità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	autorizzazioni richieste per questo caso d'uso. Vedi i ruoli IAM per AWS Data Pipeline
Consentire alle applicazioni in esecuzione su istanze Amazon EC2 di accedere alle risorse AWS	DataPipelineDefaultResourceRuolo	Creazione di un ruolo con una policy di attendibilità secondo quanto definito nella Guida per gli sviluppatori di AWS Data Pipeline	Amazon EC2 RoleforData PipelineRole

Funzione di processo per l'utente avanzato sviluppatore

AWS nome della politica gestita: Access PowerUser

Caso d'uso: questo utente esegue attività di sviluppo di applicazioni e può creare e configurare risorse e servizi che supportano lo sviluppo di applicazioni AWS consapevoli.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: la prima dichiarazione di questa politica utilizza l'`NotAction`elemento per consentire tutte le azioni per tutti i AWS servizi e per tutte le risorse tranne AWS Identity and Access Management AWS Organizations, e AWS Account Management. La seconda istruzione concede le autorizzazioni IAM per creare un ruolo collegato ai servizi. Questo è obbligatorio per alcuni servizi che devono accedere alle risorse di un altro servizio, ad esempio un bucket Amazon S3. Concede inoltre le autorizzazioni di Organizations per visualizzare le informazioni relative all'organizzazione dell'utente, tra cui l'e-mail dell'account di gestione e le limitazioni dell'organizzazione. Sebbene questa

policy limiti IAM e Organizations, consente all'utente di eseguire tutte le operazioni di IAM Identity Center se è abilitato. Concede inoltre le autorizzazioni di gestione dell'account per visualizzare quali AWS aree sono abilitate o disabilitate per l'account.

Funzione di processo per l'amministratore di rete

AWS nome della politica gestita: NetworkAdministrator

Caso d'uso: questo utente ha il compito di configurare e gestire le risorse AWS di rete.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare e gestire risorse di rete in Auto Scaling, Amazon EC2 AWS Direct Connect, Route 53, Amazon, Elastic CloudFront Load Balancing, Amazon SNS, CloudWatch Logs, AWS Elastic Beanstalk Amazon S3, IAM e Amazon Virtual CloudWatch Private Cloud. Per ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference Guide. NetworkAdministratorAWS

Questa funzione lavorativa richiede la capacità di trasferire ruoli ai AWS servizi. La policy concede `iam:GetRole` e `iam:PassRole` solo per quei ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento.

Ruoli di servizio IAM facoltativi per la funzione lavorativa di amministratore di rete

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consente ad Amazon VPC di creare e gestire i log in CloudWatch Logs per conto dell'utente per monitorare il traffico IP in entrata e in uscita dal tuo VPC	flow-logs-*	Creazione di un ruolo con una policy di attendibi lità secondo quanto definito nella Guida per l'utente di Amazon VPC	Questo caso d'uso non prevede una policy AWS gestita esistente, ma la documentazione elenca le autorizza zioni richieste. Consulta la Guida

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da selezionare
			per l'utente di Amazon VPC.

Accesso in sola lettura

AWS nome della politica gestita: Access ReadOnly

Caso d'uso: questo utente richiede l'accesso in sola lettura a tutte le risorse in un Account AWS.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della policy: questa policy concede le autorizzazioni per elencare, ottenere, descrivere e visualizzare in altro modo le risorse e i relativi attributi. Non include funzioni mutanti come create o delete. Questa politica include l'accesso in sola lettura ai AWS servizi relativi alla sicurezza, come e. AWS Identity and Access Management AWS Billing and Cost Management Visualizza la policy per l'elenco completo di servizi e operazioni supportati dalla policy.

Funzione di processo del revisore sicurezza

AWS nome della politica gestita: SecurityAudit

Caso d'uso: questo utente monitora gli account per la conformità ai requisiti di sicurezza. Questo utente può accedere ai log e agli eventi per analizzare potenziali violazioni alla sicurezza o potenziale attività non autorizzata.

Aggiornamenti della politica: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per visualizzare i dati di configurazione per molti AWS servizi e per rivederne i registri. Per ulteriori informazioni sulla politica gestita, vedere SecurityAuditnella Managed Policy Reference AWS Guide.

Funzione di processo dell'utente di Support

AWS nome della politica gestita: SupportUser

Caso d'uso: questo utente contatta l' AWS assistenza, crea casi di supporto e visualizza lo stato dei casi esistenti.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare e aggiornare AWS Support i casi. Per ulteriori informazioni sulla policy gestita, consulta SupportUserla Managed Policy Reference Guide.AWS

Funzione di processo dell'amministratore di sistema

AWS nome della politica gestita: SystemAdministrator

Caso d'uso: questo utente imposta e gestisce le risorse per le operazioni di sviluppo.

Aggiornamenti della politica: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede le autorizzazioni per creare e gestire risorse su un'ampia gamma di AWS servizi, tra cui AWS CloudTrail Amazon CloudWatch,,, AWS CodeCommit, AWS CodeDeploy AWS Config, Amazon EC2 AWS Directory Service,,, Amazon RDS AWS Identity and Access Management AWS Key Management Service AWS Lambda, Route 53, Amazon S3, Amazon SES, Amazon SQS e Amazon VPC. AWS Trusted AdvisorPer ulteriori informazioni sulla policy gestita, consulta la Managed Policy Reference Guide. SystemAdministratorAWS

Questa funzione lavorativa richiede la capacità di trasferire ruoli ai AWS servizi. La policy concede `iam:GetRole` e `iam:PassRole` solo per quei ruoli denominati nella tabella seguente. Per ulteriori informazioni, consulta Creazione dei ruoli e collegamento delle policy (console) più avanti in questo argomento. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Ruoli di servizio IAM facoltativi per la funzione lavorativa di amministratore del sistema

Caso d'uso	Nome ruolo (* è un carattere jolly)	Tipo di ruolo di servizio da selezionare	AWS politica gestita da seleziona re
Consentire alle applicazioni in esecuzione in istanze di EC2 in un cluster Amazon ECS di accedere ad Amazon ECS	ecr-sysadmin-*	Ruolo Amazon EC2 per EC2 Container Service	ruolo Amazon EC2 EC2 Container Servicefor
Consentire a un utente di monitorare i database	rds-monitoring-role	Ruolo Amazon RDS per il monitoraggio avanzato	Ruolo di Amazon EnhancedM onitoring RDS
Consenti alle app in esecuzione su istanze EC2 di accedere alle risorse. AWS	ec2-sysadmin-*	Amazon EC2	Policy di esempio per un ruolo che concede l'accesso a un bucket S3, come mostrato nella Amazon EC2 User Guide; personalizzala secondo necessità
Consenti a Lambda di leggere i flussi DynamoDB e scrivere nei log CloudWatch	lambda-sysadmin-*	AWS Lambda	AWSLambda DynamoDBE xecutionRole

Funzione di processo per utente con sola visualizzazione

AWS nome della politica gestita: Access ViewOnly

Caso d'uso: questo utente può visualizzare un elenco di AWS risorse e metadati di base dell'account in tutti i servizi. L'utente non può leggere i contenuti o i metadati delle risorse che superano la quota ed elencare informazioni per le risorse.

Aggiornamenti delle politiche: AWS mantiene e aggiorna questa politica. Per una cronologia delle modifiche apportate a questa policy, visualizza la policy nella console IAM e scegli la scheda Versioni di policy. Per ulteriori informazioni sugli aggiornamenti delle policy della funzione di processo, consulta Aggiornamenti alle politiche AWS gestite per le funzioni lavorative.

Descrizione della politica: questa politica concede`List*`, `Describe*` `Get*View*`, e `Lookup*` l'accesso alle risorse per AWS i servizi. Per vedere quali azioni include questa politica per ogni servizio, vedi ViewOnlyAccess. Per ulteriori informazioni sulla policy gestita, consulta ViewOnlyAccess in AWS Managed Policy Reference Guide.

Aggiornamenti alle politiche AWS gestite per le funzioni lavorative

Queste politiche sono tutte gestite AWS e aggiornate per includere il supporto per nuovi servizi e nuove funzionalità man mano che vengono aggiunte dai AWS servizi. Queste policy non possono essere modificate dai clienti. È possibile creare una copia della policy e quindi modificarla, ma tale copia non viene aggiornata automaticamente in quanto AWS introduce nuovi servizi e operazioni API.

Per una policy di funzione del processo, è possibile visualizzare la cronologia delle versioni e l'ora e la data di ogni aggiornamento nella console IAM. A tale scopo, utilizza i collegamenti presenti in questa pagina per visualizzare i dettagli delle policy. Quindi scegli la scheda Versioni di policy per visualizzare le versioni. Questa pagina mostra le ultime 25 versioni di una policy. Per visualizzare tutte le versioni di una policy, chiamate il AWS CLI comando get-policy-version o l'operazione Version API. GetPolicy

> **Note**
>
> È possibile avere fino a cinque versioni di una policy gestita dal cliente, ma AWS conserva la cronologia completa delle versioni delle politiche gestite. AWS

Creazione dei ruoli e collegamento delle policy (console)

Molte delle politiche elencate in precedenza concedono la possibilità di configurare AWS servizi con ruoli che consentono a tali servizi di eseguire operazioni per conto dell'utente. Le policy della funzione lavorativa specificano i nomi di ruolo esatti che è necessario utilizzare o almeno includono un prefisso che specifica la prima parte del nome che può essere utilizzato. Per creare uno di questi ruoli, eseguire le operazioni descritte nella procedura seguente.

Per creare un ruolo per una Servizio AWS (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Servizio o caso d'uso, scegli un servizio, quindi scegli il caso d'uso. I casi d'uso sono definiti dal servizio in modo da includere la policy di attendibilità richiesta dal servizio.
5. Seleziona Successivo.
6. Per i criteri di autorizzazione, le opzioni dipendono dal caso d'uso selezionato:
   - Se il servizio definisce le autorizzazioni per il ruolo, non è possibile selezionare le politiche di autorizzazione.
   - Seleziona da un set limitato di politiche di autorizzazione.
   - Seleziona tra tutte le politiche di autorizzazione.
   - Seleziona nessuna politica di autorizzazione, crea le politiche dopo la creazione del ruolo e quindi allega le politiche al ruolo.
7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.
   a. Apri la sezione Imposta i limiti delle autorizzazioni, quindi scegli Usa un limite di autorizzazioni per controllare il numero massimo di autorizzazioni per il ruolo.

      IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.
   b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
8. Seleziona Successivo.
9. Per Role name, le opzioni dipendono dal servizio:
   - Se il servizio definisce il nome del ruolo, non è possibile modificare il nome del ruolo.
   - Se il servizio definisce un prefisso per il nome del ruolo, è possibile inserire un suffisso opzionale.
   - Se il servizio non definisce il nome del ruolo, puoi assegnare un nome al ruolo.

   > **Important**
   >
   > Quando assegnate un nome a un ruolo, tenete presente quanto segue:

- I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS account e non possono essere resi unici per caso.

  Ad esempio, non creare ruoli denominati entrambi **`PRODROLE`** e**`prodrole`**. Quando un nome di ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato dai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.
- Non è possibile modificare il nome del ruolo dopo averlo creato perché altre entità potrebbero fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.
11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, nelle sezioni Passo 1: Seleziona entità attendibili o Passaggio 2: Aggiungi autorizzazioni, scegli Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consulta la sezione Applicazione di tag alle risorse IAM nella Guida per l'utente di IAM.
13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Esempio 1: configurazione di un utente come amministratore di database (console)

Questo esempio illustra i passaggi necessari per configurare Alice, un utente IAM, come Amministratore del database. Utilizza le informazioni nella prima riga della tabella nella sezione e consenti all'utente di abilitare il monitoraggio Amazon RDS. Alleghi la DatabaseAdministratorpolicy all'utente IAM di Alice in modo che possa gestire i servizi di database Amazon. Questa policy, inoltre, consente ad Alice di passare un ruolo denominato `rds-monitoring-role` al servizio Amazon RDS, che consente al servizio di monitorare i database Amazon RDS per suo conto.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Policy e inserisci **`database`** nella casella di ricerca, quindi premi Invio.
3. Seleziona il pulsante di opzione relativo alla DatabaseAdministratorpolicy, scegli Azioni, quindi scegli Allega.
4. Nell'elenco di entità, seleziona Alice, quindi scegli Collega policy. Alice ora può amministrare AWS i database. Tuttavia, per consentire ad Alice di monitorare tali database, è necessario configurare il ruolo di servizio.

5. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
6. Seleziona il tipo di ruolo Servizio AWS , quindi scegli Amazon RDS.
7. Seleziona il caso d'uso Ruolo Amazon RDS per il monitoraggio avanzato.
8. Amazon RDS definisce le autorizzazioni per il ruolo. Selezionare Next: Review (Successivo: esamina) per continuare.
9. Il nome del ruolo deve essere uno di quelli specificati dalla DatabaseAdministrator politica di cui dispone ora Alice. Uno di questi è **`rds-monitoring-role`**. Inseriscilo in Role name (Nome ruolo).
10. (Facoltativo) In Descrizione ruolo, immettere una descrizione per il nuovo ruolo.
11. Dopo aver revisionato i dettagli, selezionare Create role (Crea ruolo).
12. Alice ora può abilitare Monitoraggio avanzato RDS nella sezione Monitoraggio della console Amazon RDS. Ad esempio, può eseguire questa operazione quando crea un'istanza database, crea una replica di lettura o modifica un'istanza di database. Devono inserire il nome del ruolo che hanno creato (rds-monitoring-role) nella casella Ruolo di monitoraggio quando impostano Abilita monitoraggio avanzato su Sì.

Esempio 2: configurazione di un utente come amministratore di rete (console)

Questo esempio illustra i passaggi necessari per configurare Jorge, un utente IAM, come Amministratore di rete. Utilizza le informazioni nella tabella in tale sezione per consentire a Jorge di monitorare il traffico IP in uscita e in entrata da VPC. Consente inoltre a Jorge di acquisire tali informazioni nei log in CloudWatch Logs. Alleghi la NetworkAdministratorpolicy all'utente IAM di Jorge in modo che possa configurare le risorse di rete. AWS Inoltre, tale policy consente a Jorge di passare un ruolo il cui nome inizia con `flow-logs*` ad Amazon EC2 al momento della creazione del log di flusso. In questo scenario, diversamente dall'esempio 1, non è disponibile un tipo di ruolo di servizio predefinito, è necessario eseguire pochi passaggi in maniera diversa.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy e inserisci **`network`** nella casella di ricerca, quindi premi Invio.
3. Seleziona il pulsante di opzione accanto alla NetworkAdministratorpolicy, scegli Azioni, quindi scegli Allega.

4. Nell'elenco degli utenti, seleziona la casella di controllo accanto a Jorge e scegli Attach policy (Collega policy). Jorge ora può amministrare le risorse di AWS rete. Tuttavia, per consentire il monitoraggio di traffico IP nel VPC, è necessario configurare il ruolo del servizio.
5. Poiché il ruolo del servizio che bisogna creare non dispone di una policy gestita predefinita, è necessario prima crearlo. Nel riquadro di navigazione, selezionare Policies (Policy) e Create Policy (Crea policy).
6. Nella sezione Editor di policy, seleziona l'opzione JSON e copia il testo dal seguente documento della policy JSON. Incolla il testo nella casella di testo JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

7. Risolvi eventuali avvisi di sicurezza, errori o avvisi generali generati durante la convalida delle policy, quindi scegli Next (Successivo).

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy.

8. Nella pagina Verifica e crea, digita **`vpc-flow-logs-policy-for-service-role`** come nome della policy. Rivedi il campo Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy, quindi seleziona Crea policy per salvare il lavoro.

La nuova policy appare nell'elenco delle policy gestite ed è pronta a collegare.

9. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
10. Seleziona il tipo di ruolo Servizio AWS , quindi scegli Amazon EC2.
11. Seleziona il caso d'uso Amazon EC2.
12. Nella pagina Allega criteri di autorizzazione, scegli la politica che hai creato in precedenza, vpc-flow-logs-policy- for-service-role, quindi scegli Avanti: revisione.
13. Il nome del ruolo deve essere consentito dalla NetworkAdministrator politica attuale di Jorge. Qualsiasi nome che inizia con `flow-logs-` è consentito. Per questo esempio, inserisci **`flow-logs-for-jorge`** come Role name (Nome del ruolo).
14. (Facoltativo) In Descrizione ruolo, immettere una descrizione per il nuovo ruolo.
15. Dopo aver revisionato i dettagli, selezionare Create role (Crea ruolo).
16. Ora è possibile configurare la policy attendibilità necessaria per questo scenario. Nella pagina Ruoli, scegli il flow-logs-for-jorgeruolo (il nome, non la casella di controllo). Nella pagina dei dettagli per il nuovo ruolo, selezionare la scheda Trust relationships (Relazioni di trust) e selezionare Edit trust relationship (Modifica relazione di trust).
17. Modificare la riga "Servizio" come segue, sostituendo la voce per `ec2.amazonaws.com`:

```
        "Service": "vpc-flow-logs.amazonaws.com"
```

18. Jorge può ora creare log di flusso da un VPC o una sottorete nella console Amazon EC2. Quando crei il log di flusso, specifica il flow-logs-for-jorgeruolo. Quel ruolo dispone delle autorizzazioni per creare il log e scriverci dati.

AWS chiavi di contesto della condizione globale

Quando un principale effettua una richiesta a AWS, AWS raccoglie le informazioni sulla richiesta in un contesto di richiesta. È possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi della richiesta con i valori chiave specificati nella policy. Le informazioni sulla richiesta vengono fornite da diverse fonti, tra cui il responsabile che effettua la richiesta, la risorsa sulla quale viene effettuata la richiesta e i metadati relativi alla richiesta stessa.

Le chiavi di condizione globali possono essere utilizzate in tutti i AWS servizi. Sebbene queste chiavi di condizione possano essere utilizzate in tutte le politiche, la chiave non è disponibile in tutti i contesti di richiesta. Ad esempio, la chiave di `aws:SourceAccount` condizione è disponibile solo

quando la chiamata alla risorsa viene effettuata direttamente da un responsabile AWS del servizio. Per ulteriori informazioni sulle circostanze in cui una chiave globale viene inclusa nel contesto della richiesta, consulta le informazioni sulla disponibilità per ciascuna chiave.

Alcuni servizi individuali creano le proprie chiavi di condizione che sono disponibili nel contesto della richiesta per altri servizi. Le chiavi di condizione tra servizi sono un tipo di chiave di condizione globale che includono un prefisso corrispondente al nome del servizio, ad esempio `ec2:` `olambda:`, ma sono disponibili in altri servizi.

Le chiavi di condizione specifiche del servizio sono definite per l'uso con un singolo servizio. AWS Ad esempio, Amazon S3 consente di scrivere una policy con la chiave di `s3:VersionId` condizione per limitare l'accesso a una versione specifica di un oggetto Amazon S3. Questa chiave di condizione è unica per il servizio, il che significa che funziona solo con le richieste al servizio Amazon S3. Per le chiavi di condizione specifiche del servizio, consulta Actions, Resources e Condition Keys for AWS Services e scegli il servizio di cui desideri visualizzare le chiavi.

> **Note**
>
> Se si utilizzano chiavi di condizione disponibili solo in alcune circostanze, è possibile utilizzare IfExistsle versioni degli operatori di condizione. Se le chiavi di condizione sono assenti da un contesto di richiesta, la policy può non riuscire a effettuare la valutazione. Ad esempio, utilizzare il seguente blocco condizionale con gli operatori `...IfExists` per verificare se una richiesta proviene da uno specifico intervallo IP o da un determinato VPC. Se una o entrambe le chiavi non sono incluse nel contesto della richiesta, la condizione restituisce comunque `true`. I valori vengono controllati solo se la chiave specificata è inclusa nel contesto della richiesta. Per ulteriori informazioni su come viene valutata una politica quando una chiave non è presente per altri operatori, consulta Operatori di condizione.
>
> ```
> "Condition": {
>     "IpAddressIfExists": {"aws:SourceIp" : ["xxx"] },
>     "StringEqualsIfExists" : {"aws:SourceVpc" : ["yyy"]}
> }
> ```

> **Important**
>
> Per confrontare la condizione con un contesto di richiesta con più valori chiave, devi utilizzare gli operatori su set `ForAllValues` o `ForAnyValue`. Utilizza gli operatori di insieme solo

con le chiavi della condizione multivalore. Non utilizzare operatori con chiavi di condizione a valore singolo. Per ulteriori informazioni, consulta Chiavi di contesto multivalore.

Proprietà del principale	Proprietà di una sessione di ruolo	Proprietà della rete	Le proprietà della risorsa	Proprietà della richiesta
Leggi: PrincipalArn leggi: PrincipalAccount aws: PrincipalOrg Percorsi aws: PrincipalOrg ID aws:PrincipalTag//tag-key leggi: PrincipalIsAWSService aws: PrincipalService Nome leggi: PrincipalServiceNamesList leggi: PrincipalType aws:userid aws:username	leggi: FederatedProvider aws: TokenIssueOra leggi: MultiFactorAuthAge leggi: MultiFactorAuthPresent AWS: EC2 VpcInstanceSource AWS: EC2 IPv4 privatoInstanceSource leggi: SourceIdentity ec2: RoleDelivery ec2: ArnSourceInstance colla: RoleAssumed Di	leggi: SourceIp come: SourceVpc leggi: SourceVpce aws: VpcSourceIp	Leggi: ResourceAccount aws: ResourceOrg Percorsi Leggi: ResourceOrg ID aws:ResourceTag//tag-key	Leggi: CalledVia aws: CalledViaPrimo aws: CalledViaUltimo AWS: via AWSService leggi: CurrentTime leggi: EpochTime aws:Referer Leggi: RequestedRegion aws:RequestTag//tag-key leggi: TagKeys leggi: SecureTransport leggi: SourceArn

Proprietà del principale	Proprietà di una sessione di ruolo	Proprietà della rete	Le proprietà della risorsa	Proprietà della richiesta
	glue: Servizio CredentialIssuing lambda: Arn SourceFunction ssm: Arn SourceInstance archivio di identità: UserId			leggi: SourceAccount aws: SourceOrg Percorsi aws: SourceOrg ID leggi: UserAgent

Proprietà del principale

Utilizzate le seguenti chiavi di condizione per confrontare i dettagli sul principale che effettua la richiesta con le proprietà principali specificate nella politica. Per un elenco dei responsabili che possono effettuare richieste, vedereSpecifica di un'entità principale.

Indice

Leggi: PrincipalArn

Utilizzare questa chiave per confrontare il nome della risorsa Amazon (ARN) del principale che ha effettuato la richiesta con l'ARN specificato nella policy. Per i ruoli IAM, il contesto della richiesta restituisce l'ARN del ruolo, non l'ARN dell'utente che ha assunto il ruolo.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste firmate. Le richieste anonime non includono questa chiave. È possibile specificare i seguenti tipi di principali in questa chiave di condizione:
  - Ruolo IAM
  - Utente IAM
  - AWS STS sessione utente federata
  - Account AWS utente root
- Tipo di dati: ARN, String

  AWS consiglia di utilizzare operatori ARN anziché operatori stringa quando si confrontano gli ARN.
- Tipo di valore: valore singolo
- Valori di esempio L'elenco seguente mostra il valore del contesto della richiesta restituito per diversi tipi di principali che è possibile specificare nella `aws:PrincipalArn` chiave di condizione:
  - Ruolo IAM- Il contesto della richiesta contiene il seguente valore per la chiave di condizione `aws:PrincipalArn`. Non specificare l'ARN della sessione del ruolo assunto come valore per questa chiave di condizione. Per ulteriori informazioni sul principale della sessione del ruolo assunto, consulta Principali della sessione come ruolo.

    ```
    arn:aws:iam::123456789012:role/role-name
    ```
  - Utente IAM- Il contesto della richiesta contiene il seguente valore per la chiave di condizione `aws:PrincipalArn`.

    ```
    arn:aws:iam::123456789012:user/user-name
    ```
  - AWS STS sessioni utente federate: il contesto della richiesta contiene il seguente valore per la chiave di condizione. `aws:PrincipalArn`

    ```
    arn:aws:sts::123456789012:federated-user/user-name
    ```

- Account AWS utente root — Il contesto della richiesta contiene il seguente valore per la chiave `aws:PrincipalArn` di condizione. Quando si specifica l'ARN dell'utente root come valore per la chiave di condizione `aws:PrincipalArn`, limita le autorizzazioni solo per l'utente root del Account AWS. Questo è diverso dallo specificare l'ARN dell'utente root nell'elemento principale di una policy basata sulle risorse, che delega l'autorità al Account AWS. Per ulteriori informazioni sulla specifica dell'ARN dell'utente root nell'elemento principale di una policy basata sulle risorse, consulta Account AWS presidi.

  ```
  arn:aws:iam::123456789012:root
  ```

È possibile specificare l'ARN dell'utente root come valore per la chiave di condizione `aws:PrincipalArn` nelle politiche di controllo AWS Organizations del servizio (SCP). Le SPC sono un tipo di policy dell'organizzazione utilizzate per gestire le autorizzazioni nell'organizzazione e riguardano solo gli account dei membri nell'organizzazione. Una SCP limita le autorizzazioni per i ruoli e gli utenti IAM e negli account membri, compreso l'utente root dell'account membro. Per ulteriori informazioni sull'effetto delle SCP sulle autorizzazioni, consulta Effetti delle SCP sulle autorizzazioni nella Guida per l'utente di Organizations.

leggi: PrincipalAccount

Utilizzare questa chiave per confrontare l'account a cui appartiene il principale richiedente con l'identificatore dell'account specificato nella policy. Per le richieste anonime, il contesto della richiesta restituisce `anonymous`.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Nell'esempio seguente, l'accesso è negato tranne che ai principali con il numero di account `123456789012`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessFromPrincipalNotInSpecificAccount",
```

```
      "Action": "service:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:service:region:accountID:resource"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

aws: PrincipalOrg Percorsi

Usa questa chiave per confrontare il AWS Organizations percorso del principale che effettua la richiesta con il percorso indicato nella policy. Tale principale può essere un utente IAM, un ruolo IAM, un utente federato o Utente root dell'account AWS. In una policy, questa chiave di condizione garantisce che il richiedente sia un membro dell'account all'interno della radice dell'organizzazione specificata o delle unità organizzative (OU) in AWS Organizations. Un AWS Organizations percorso è una rappresentazione testuale della struttura di un'entità Organizations. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, consultare Informazioni sul percorso dell'entità AWS Organizations.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se il principale è membro di un'organizzazione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

> **Note**
>
> Gli ID organizzazione sono univoci a livello globale, ma gli ID delle unità organizzative e gli ID root sono univoci solo all'interno di un'organizzazione. Ciò significa che non ci sono due organizzazioni che condividono lo stesso ID organizzazione. Tuttavia, un'altra organizzazione

potrebbe avere un'unità organizzativa o un root con il tuo stesso ID. Si consiglia di includere sempre l'ID organizzazione quando si specifica un'unità organizzativa o un root.

Ad esempio, la seguente condizione restituisce `true` per le entità negli account collegati direttamente all'unità organizzativa `ou-ab12-22222222`, ma non nelle unità organizzative figlie.

```
"Condition" : { "ForAnyValue:StringEquals" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/"]
}}
```

La seguente condizione restituisce `true` per i principali in un account collegato direttamente all'unità organizzativa o a una delle sue unità organizzative figlie. Quando si include un carattere jolly, è necessario utilizzare l'operatore condizionale `StringLike`.

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/
*"]
}}
```

La seguente condizione restituisce `true` per i principali in un account collegato direttamente a una delle unità organizzative figlie ma non all'unità organizzativa padre. La condizione precedente è per l'unità organizzativa o per qualsiasi figlio. La condizione seguente è solo per i figli (e tutti i figli di quei figli).

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/
ou-*"]
}}
```

La condizione seguente consente l'accesso per ogni principale nell'organizzazione `o-a1b2c3d4e5`, indipendentemente dalla propria unità organizzativa padre.

```
"Condition" : { "ForAnyValue:StringLike" : {
     "aws:PrincipalOrgPaths":["o-a1b2c3d4e5/*"]
}}
```

`aws:PrincipalOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. Quando si utilizzano più valori con l'operatore

condizionale `ForAnyValue`, il percorso del principale deve corrispondere a uno dei percorsi elencati nella policy. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Chiavi di contesto multivalore.

```
    "Condition": {
        "ForAnyValue:StringLike": {
            "aws:PrincipalOrgPaths": [
                "o-a1b2c3d4e5/r-ab12/ou-ab12-33333333/*",
                "o-a1b2c3d4e5/r-ab12/ou-ab12-22222222/*"
            ]
        }
    }
```

aws: PrincipalOrg ID

Utilizza questa chiave per confrontare l'identificatore dell'organizzazione AWS Organizations a cui appartiene il principale richiedente con l'identificatore specificato nella politica.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se il principale è membro di un'organizzazione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Questa chiave globale fornisce un'alternativa per elencare tutti gli ID account per tutti gli account AWS all'interno di un'organizzazione. È possibile utilizzare questa chiave di condizione per specificare semplicemente l'elemento `Principal` in una policy basata sulle risorse. È possibile specificare l'ID organizzazione nell'elemento condizionale. Quando si aggiunge e si rimuove un account, le policy che includono la chiave `aws:PrincipalOrgID` includono automaticamente anche gli account corretti e non necessitano di aggiornamento manuale.

Ad esempio, la seguente policy del bucket Amazon S3 consente ai membri di qualsiasi account nell'organizzazione `o-xxxxxxxxxxx` di aggiungere un oggetto al bucket `policy-ninja-dev`.

```
{
 "Version": "2012-10-17",
 "Statement": {
   "Sid": "AllowPutObject",
   "Effect": "Allow",
   "Principal": "*",
   "Action": "s3:PutObject",
```

```
    "Resource": "arn:aws:s3:::policy-ninja-dev/*",
    "Condition": {"StringEquals":
      {"aws:PrincipalOrgID":"o-xxxxxxxxxxx"}
    }
  }
}
```

> **Note**
>
> Questa condizione globale vale anche per l'account di gestione di un'organizzazione AWS . Questa policy impedisce a tutti principali esterni all'organizzazione specificata di accedere al bucket Amazon S3. Sono inclusi tutti AWS i servizi che interagiscono con le risorse interne, come AWS CloudTrail l'invio di dati di log ai bucket Amazon S3. Per scoprire come concedere l'accesso ai AWS servizi in modo sicuro, consulta. leggi: PrincipalIs AWSService

Per ulteriori informazioni su AWS Organizations, consulta What Is AWS Organizations? nella Guida AWS Organizations per l'utente.

aws:PrincipalTag//tag-key

Utilizzare questa chiave per confrontare il tag collegato al principale che effettua la richiesta con il tag specificato nella policy. Se il principale ha più di un tag collegato, il contesto della richiesta include una chiave `aws:PrincipalTag` per ogni chiave tag collegata.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta se il principale utilizza un utente IAM con tag collegati. È inclusa per un principale che utilizza un ruolo IAM con tag collegati o tag di sessione. Le richieste anonime non includono questa chiave.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

È possibile aggiungere attributi personalizzati a un utente o a un ruolo sotto forma di coppia chiave-valore. Per ulteriori informazioni sui tag in IAM, consulta Tagging delle risorse IAM. Puoi utilizzare `aws:PrincipalTag` per controllare l'accesso per i principali AWS .

Questo esempio mostra come creare una policy basata sull'identità che consenta agli utenti con il tag **`department=hr`** per gestire utenti, gruppi o ruoli IAM. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/department": "hr"
        }
      }
    }
  ]
}
```

leggi: PrincipalIs AWSService

Usa questa chiave per verificare se la chiamata alla tua risorsa viene effettuata direttamente da un responsabile AWS del servizio. Ad esempio, AWS CloudTrail utilizza il principale del servizio `cloudtrail.amazonaws.com` per scrivere log nel bucket Amazon S3. La chiave di contesto della richiesta è impostata su true (VERO) quando un servizio utilizza un principale del servizio per eseguire un'operazione diretta sulle risorse. La chiave di contesto è impostata su false se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale. Viene impostata su false anche se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.

- Disponibilità: questa chiave è presente nel contesto della richiesta per tutte le richieste di API firmate che utilizzano le credenziali AWS . Le richieste anonime non includono questa chiave.
- Tipo di dati: booleano
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per limitare l'accesso alle identità attendibili e alle posizioni di rete previste, garantendo al contempo l'accesso ai servizi in modo sicuro. AWS

Nel seguente esempio di policy sui bucket di Amazon S3, l'accesso al bucket è limitato a meno che la richiesta non provenga da `vpc-111bbb22` o provenga da un responsabile del servizio, ad esempio. CloudTrail

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Expected-network+service-principal",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/AWS Logs/AccountNumber/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-111bbb22"
        },
        "BoolIfExists": {
          "aws:PrincipalIsAWSService": "false"
        }
      }
    }
  ]
}
```

Nel seguente video, scopri ulteriori informazioni su come utilizzare la chiave di condizione `aws:PrincipalIsAWSService` in una policy.

Concedi in modo sicuro l'accesso congiunto agli utenti autorizzati, alle posizioni di rete previste e ai servizi. AWS

aws: PrincipalService Nome

Utilizza questa chiave per confrontare il nome del principale del servizio nella policy con il prinicpale del servizio che effettua richieste alle risorse. È possibile utilizzare questa chiave per verificare se la chiamata viene effettuata da un principale del servizio specifico. Quando un principale del servizio effettua una richiesta diretta alla risorsa, la chiave `aws:PrincipalServiceName` contiene il nome del principale del servizio. Ad esempio, il nome principale del AWS CloudTrail servizio è`cloudtrail.amazonaws.com`.

- Disponibilità: questa chiave è presente nella richiesta quando la chiamata viene effettuata da un responsabile AWS del servizio. Questa chiave non è presente in alcun'altra situazione, tra cui:
  - Se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.

  - Se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale.
  - Se la chiamata viene effettuata direttamente da un principale IAM.
  - Se la chiamata viene effettuata da un richiedente anonimo.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per limitare l'accesso alle identità attendibili e alle posizioni di rete previste, garantendo al contempo l'accesso a un AWS servizio in tutta sicurezza.

Nel seguente esempio di policy sui bucket di Amazon S3, l'accesso al bucket è limitato a meno che la richiesta non provenga da `vpc-111bbb22` o provenga da un responsabile del servizio, ad esempio. CloudTrail

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "expected-network+service-principal",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/AWS Logs/AccountNumber/*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-111bbb22",
          "aws:PrincipalServiceName": "cloudtrail.amazonaws.com"
        }
      }
    }
  ]
}
```

leggi: PrincipalService NamesList

Questa chiave fornisce un elenco di tutti i nomi dei principali del servizio che appartengono al servizio. Questa è una chiave di condizione avanzata. È possibile utilizzarlo per impedire al servizio di accedere alla risorsa solo da una regione specifica. Alcuni servizi possono creare entità di servizio

regionali per indicare una particolare istanza del servizio all'interno di una regione specifica. È possibile limitare l'accesso a una risorsa a una particolare istanza del servizio. Quando un principale del servizio effettua una richiesta diretta alla risorsa, `aws:PrincipalServiceNamesList` contiene un elenco non ordinato di tutti i nomi di principali del servizio associati all'istanza regionale del servizio.

- Disponibilità: questa chiave è presente nella richiesta quando la chiamata viene effettuata da un responsabile AWS del servizio. Questa chiave non è presente in alcun'altra situazione, tra cui:
  - Se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale.
  - Se il servizio utilizza le credenziali di un principale IAM per effettuare una richiesta per conto del principale.
  - Se la chiamata viene effettuata direttamente da un principale IAM.
  - Se la chiamata viene effettuata da un richiedente anonimo.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

`aws:PrincipalServiceNamesList` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Chiavi di contesto multivalore.

leggi: PrincipalType

Utilizzare questa chiave per confrontare il tipo di principale che effettua la richiesta con il tipo di principale specificato nella policy. Per ulteriori informazioni, consulta Specifica di un'entità principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

aws:userid

Utilizzare questa chiave per confrontare l'identificatore del principale richiedente con l'ID specificato nella policy. Per gli utenti IAM, il valore del contesto della richiesta è l'ID utente. Per i ruoli IAM, questo formato di valore può variare. Per informazioni dettagliate su come vengono visualizzate le informazioni per diverse entità, consultare Specifica di un'entità principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta per tutte le richieste, incluse le richieste anonime.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

aws:username

Utilizzare questa chiave per confrontare il nome utente del richiedente con il nome utente specificato nella policy. Per informazioni dettagliate su come vengono visualizzate le informazioni per diverse entità, consultare Specifica di un'entità principale. Per esempi specifici di valori chiave `principal`, vedi Valori della chiave dell'entità principale.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta per gli utenti IAM. Le richieste anonime e le richieste effettuate utilizzando i ruoli Utente root dell'account AWS o IAM non includono questa chiave. Le richieste effettuate utilizzando le credenziali di IAM Identity Center non includono questa chiave nel contesto.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Proprietà di una sessione di ruolo

Utilizzate le seguenti chiavi di condizione per confrontare le proprietà della sessione di ruolo al momento della generazione della sessione. Queste chiavi di condizione sono disponibili solo quando una richiesta viene effettuata da un principale con credenziali di sessione di ruolo o utente federato. I valori di queste chiavi di condizione sono incorporati nel token di sessione del ruolo.

Un ruolo è un tipo di principale. È inoltre possibile utilizzare le chiavi di condizione della Proprietà del principale sezione per valutare le proprietà di un ruolo quando un ruolo effettua una richiesta.

Indice

- leggi: FederatedProvider
- aws: TokenIssue Ora
- leggi: MultiFactor AuthAge
- leggi: MultiFactor AuthPresent
- AWS: EC2 Vpc InstanceSource
- AWS: EC2 IPv4 privato InstanceSource
- leggi: SourceIdentity
- ec2: RoleDelivery
- ec2: Arn SourceInstance
- colla: RoleAssumed Di
- glue: Servizio CredentialIssuing
- lambda: Arn SourceFunction
- ssm: Arn SourceInstance
- archivio di identità: UserId

leggi: FederatedProvider

Utilizzare questa chiave per confrontare il provider dell'identità di emissione (IdP) del principale con l'IdP specificato nella policy. Ciò significa che è stato assunto un ruolo IAM utilizzando l'`AssumeRoleWithWebIdentity` AWS STS operazione. Quando le credenziali temporanee della sessione come ruolo risultante vengono utilizzate per effettuare una richiesta, il contesto della richiesta identifica l'IdP che ha autenticato l'identità federata originale.

- Disponibilità: questa chiave è presente quando il principale è un principale della sessione come ruolo e quella sessione è stata emessa quando un ruolo è stato assunto con `AssumeRoleWithWebIdentity`.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Ad esempio, se l'utente è stato autenticato tramite Amazon Cognito, il contesto delle richiesta include il valore `cognito-identity.amazonaws.com`. Analogamente, se l'utente è stato autenticato tramite Login with Amazon, il contesto della richiesta contiene il valore `www.amazon.com`.

È possibile utilizzare qualsiasi chiave di condizione a valore singolo come variabile. L'esempio seguente di policy basata sulle risorse utilizza la chiave `aws:FederatedProvider` come variabile di policy nell'ARN di una risorsa. Questa policy consente a qualsiasi principale che ha effettuato l'autenticazione tramite un IdP di estrarre oggetti da un bucket Amazon S3 con un percorso specifico per il provider di identità di emissione.

aws: TokenIssue Ora

Utilizzare questa chiave per confrontare la data e l'ora in cui sono state emesse le credenziali di sicurezza temporanee con la data e l'ora specificate nella policy.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali temporanee per effettuare la richiesta. La chiave non è presente nelle AWS CLI richieste AWS API o AWS SDK effettuate utilizzando le chiavi di accesso.
- Tipo di dati: data
- Tipo di valore: valore singolo

Per sapere quali servizi supportano l'utilizzo di credenziali temporanee, consulta AWS servizi che funzionano con IAM.

leggi: MultiFactor AuthAge

Utilizzare questa chiave per confrontare il numero di secondi da quando il principale richiedente è stato autorizzato utilizzando MFA con il numero specificato nella policy. Per ulteriori informazioni sulla funzionalità MFA, consultare Utilizzo dell'autenticazione a più fattori (MFA) in AWS.

> **Important**
>
> Questa chiave condizionale non è presente per le identità federate o le richieste effettuate utilizzando chiavi di accesso per firmare richieste AWS CLI, AWS API o SDK. AWS Per ulteriori informazioni sull'aggiunta della protezione MFA alle operazioni API con credenziali di sicurezza temporanee, consulta. Configurazione dell'accesso alle API protetto da MFA
> Per verificare se l'autenticazione MFA viene utilizzata per convalidare le identità federate IAM, puoi passare il metodo di autenticazione dal tuo provider di identità come tag di AWS sessione. Per informazioni dettagliate, vedi Passare i tag di sessione AWS STS.
> Per applicare la MFA per le identità di IAM Identity Center, puoi abilitare gli attributi per il controllo degli accessi per passare una dichiarazione di asserzione SAML con il metodo di autenticazione dal tuo provider di identità a IAM Identity Center.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali di sicurezza temporanee per effettuare la richiesta. Le politiche con condizioni MFA possono essere allegate a:
  - Un utente o un gruppo IAM
  - Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
  - La policy di attendibilità di un ruolo IAM che può essere assunto da un utente
- Tipo di dati: numerico
- Tipo di valore: valore singolo

leggi: MultiFactor AuthPresent

Utilizza questa chiave per verificare se è stata utilizzata l'autenticazione a più fattori (MFA) per convalidare le credenziali di sicurezza temporanee che hanno effettuato la richiesta.

> **Important**
>
> Questa chiave condizionale non è presente per le identità federate o le richieste effettuate utilizzando chiavi di accesso per firmare richieste AWS CLI, AWS API o SDK. AWS Per ulteriori informazioni sull'aggiunta della protezione MFA alle operazioni API con credenziali di sicurezza temporanee, consulta. Configurazione dell'accesso alle API protetto da MFA
> Per verificare se l'autenticazione MFA viene utilizzata per convalidare le identità federate IAM, puoi passare il metodo di autenticazione dal tuo provider di identità come tag di AWS sessione. Per informazioni dettagliate, vedi Passare i tag di sessione AWS STS.
> Per applicare la MFA per le identità di IAM Identity Center, puoi abilitare gli attributi per il controllo degli accessi per passare una dichiarazione di asserzione SAML con il metodo di autenticazione dal tuo provider di identità a IAM Identity Center.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando il principale utilizza credenziali temporanee per effettuare la richiesta. Le politiche con condizioni MFA possono essere allegate a:
  - Un utente o un gruppo IAM
  - Una risorsa, ad esempio un bucket Amazon S3, una coda Amazon SQS o un argomento Amazon SNS
  - La policy di attendibilità di un ruolo IAM che può essere assunto da un utente

- Tipo di dati: booleano
- Tipo di valore: valore singolo

Le credenziali temporanee vengono utilizzate per autenticare i ruoli IAM e gli utenti IAM con token temporanei di AssumeRoleor GetSessionToken e gli utenti di. AWS Management Console

Le chiavi di accesso utente IAM sono credenziali a lungo termine, ma in alcuni casi AWS creano credenziali temporanee per conto degli utenti IAM per eseguire operazioni. In questi casi, la chiave `aws:MultiFactorAuthPresent` è presente nella richiesta ed è impostata su un valore `false`. Esistono due casi comuni in cui ciò può accadere:

- Gli utenti IAM utilizzano AWS Management Console inconsapevolmente credenziali temporanee. Gli utenti accedono alla console utilizzando il nome utente e la password, che sono credenziali a lungo termine. Tuttavia, in background, la console genera credenziali temporanee per conto dell'utente.
- Se un utente IAM effettua una chiamata a un AWS servizio, il servizio riutilizza le credenziali dell'utente per effettuare un'altra richiesta a un servizio diverso. Ad esempio, quando si chiama Athena per accedere a un bucket Amazon S3 o quando lo si AWS CloudFormation utilizza per creare un'istanza Amazon EC2. Per la richiesta successiva, utilizza credenziali temporanee AWS .

Per sapere quali servizi supportano l'utilizzo di credenziali temporanee, consulta AWS servizi che funzionano con IAM.

La chiave `aws:MultiFactorAuthPresent` non è presente quando vengono lanciati un'API o un comando della CLI con credenziali a lungo termine, ad esempio coppie di chiavi di accesso. Pertanto, si consiglia, quando si controlla questa chiave, di utilizzare le versioni `...IfExists` degli operatori di condizione.

È importante capire che il seguente elemento `Condition` non è un modo affidabile per controllare se una richiesta è autenticata utilizzando MFA.

```
#####   WARNING: NOT RECOMMENDED   #####
"Effect" : "Deny",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di effetto `Deny`, elemento `Bool` e valore `false` nega le richieste che possono essere autenticate utilizzando MFA, ma che non lo sono. Ciò è valido solo per le credenziali

temporanee che supportano l'uso di MFA. Questa istruzione non nega l'accesso alle richieste effettuate utilizzando le credenziali a lungo termine oppure alle richieste autenticate utilizzando MFA. Utilizza questo esempio con cautela in quanto la relativa logica è complessa e non verifica se l'autenticazione MFA è stata effettivamente utilizzata.

Inoltre, non utilizzare la combinazione di effetto `Deny`, elemento `Null` e `true` perché ha lo stesso comportamento e la logica è ancora più complessa.

Combinazione consigliata

Consigliamo di utilizzare l'operatore `BoolIfExists` per verificare se una richiesta viene autenticata con MFA.

```
"Effect" : "Deny",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di `Deny`, `BoolIfExists` e `false` nega le richieste che non vengono autenticate con MFA. Nello specifico, nega le richieste da credenziali temporanee che non includono MFA. Nega inoltre le richieste effettuate utilizzando credenziali a lungo termine, ad esempio AWS CLI operazioni AWS API effettuate utilizzando chiavi di accesso. L'operatore `*IfExists` verifica la presenza della chiave `aws:MultiFactorAuthPresent` e se potrebbe essere presente o meno, come indicato dalla relativa esistenza. Utilizzalo quando intendi negare qualsiasi richiesta non autenticata con MFA. È più sicuro, ma può violare qualsiasi codice o script che utilizza le chiavi di accesso per accedere all' AWS CLI API or. AWS

Combinazioni alternative

È inoltre possibile utilizzare l'`BoolIfExists`operatore per consentire le richieste autenticate tramite MFA AWS CLI e/o le richieste AWS API effettuate utilizzando credenziali a lungo termine.

```
"Effect" : "Allow",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : "true" } }
```

Questa condizione è corrispondente se la chiave esiste ed è presente o se la chiave non esiste. Questa combinazione di `Allow`, `BoolIfExists` e `true` consente le richieste autenticate con MFA o le richieste che non possono essere autenticate con MFA. Ciò significa che le AWS CLI operazioni AWS API e AWS SDK sono consentite quando il richiedente utilizza le proprie chiavi di accesso a lungo termine. Questa combinazione non consente le richieste da credenziali temporanee che potrebbero ma non includono MFA.

Quando crei una policy utilizzando l'editor visivo della console IAM e scegli MFA obbligatoria, questa combinazione viene applicata. Questa impostazione richiede MFA per l'accesso alla console, ma consente l'accesso programmatico senza MFA.

In alternativa, puoi utilizzare l'operatore `Bool` per consentire le richieste programmatiche e della console solo quando autenticate tramite MFA.

```
"Effect" : "Allow",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : "true" } }
```

Questa combinazione di `Allow`, `Bool` e `true` consente solo le richieste autenticate mediante MFA. Ciò è valido solo per le credenziali temporanee che supportano l'uso di MFA. Questa istruzione non consente l'accesso alle richieste eseguite utilizzando chiavi di accesso a lungo termine oppure alle richieste eseguite utilizzando credenziali temporanee senza MFA.

Non utilizzare una struttura di policy simile alle seguenti per controllare se la chiave MFA è presente:

```
#####   WARNING: USE WITH CAUTION   #####

"Effect" : "Allow",
"Condition" : { "Null" : { "aws:MultiFactorAuthPresent" : "false" } }
```

Questa combinazione di effetto `Allow`, elemento `Null` e valore `false` consente solo le richieste che possono essere autenticate mediante MFA, indipendentemente dall'autenticazione o meno della richiesta. Ciò consente tutte le richieste eseguite utilizzando credenziali temporanee e nega l'accesso alle credenziali a lungo termine. Utilizza questo esempio con cautela in quanto non verifica se l'autenticazione MFA è stata effettivamente utilizzata.

AWS: EC2 Vpc InstanceSource

Questa chiave identifica il VPC al quale sono state consegnate le credenziali del ruolo IAM di Amazon EC2. È possibile utilizzare questa chiave in una policy con la chiave globale `aws:SourceVPC` per verificare se da un VPC (`aws:SourceVPC`) viene effettuata una chiamata che corrisponde al VPC al quale è stata consegnata una credenziale (`aws:Ec2InstanceSourceVpc`).

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che il richiedente firma le richieste con una credenziale di ruolo Amazon EC2. Può essere utilizzata nelle policy IAM, nelle policy di controllo dei servizi, nelle policy degli endpoint VPC e nelle policy delle risorse.
- Tipo di dati: stringa

- Tipo di valore: valore singolo

Questa chiave può essere utilizzata con i valori identificativi del VPC, ma è particolarmente utile se impiegata come variabile in combinazione con la chiave di contesto `aws:SourceVpc`. La chiave di contesto `aws:SourceVpc` viene inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta. L'impiego di `aws:Ec2InstanceSourceVpc` con `aws:SourceVpc` consente di utilizzare `aws:Ec2InstanceSourceVpc` in modo più ampio, poiché confronta dei valori che in genere cambiano insieme.

> **Note**
>
> Questa chiave di condizione non è disponibile in EC2-Classic.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireSameVPC",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
            "aws:SourceVpc": "${aws:Ec2InstanceSourceVpc}"
        },
        "Null": {
          "ec2:SourceInstanceARN": "false"
        },
        "BoolIfExists": {
          "aws:ViaAWSService": "false"
        }
      }
    }
  ]
}
```

Nell'esempio precedente, l'accesso è negato se il valore di `aws:SourceVpc` non corrisponde al valore di `aws:Ec2InstanceSourceVpc`. L'istruzione della policy è limitata ai soli ruoli

utilizzati come ruoli dell'istanza Amazon EC2 verificando l'esistenza della chiave di condizione `ec2:SourceInstanceARN`.

La policy consente `aws:ViaAWSService` di AWS autorizzare le richieste quando vengono effettuate per conto dei ruoli delle istanze Amazon EC2. Ad esempio, quando effettui una richiesta da un'istanza Amazon EC2 a un bucket Amazon S3 crittografato, Amazon S3 effettua una chiamata a per tuo conto. AWS KMS Alcune chiavi non sono presenti quando viene effettuata la richiesta a. AWS KMS

AWS: EC2 IPv4 privato InstanceSource

Questa chiave identifica l'indirizzo IPv4 privato dell'interfaccia di rete elastica primaria a cui sono state consegnate le credenziali del ruolo IAM di Amazon EC2. Per assicurarti di disporre di una combinazione unica a livello globale di ID VPC e IP privato di origine, devi utilizzare questa chiave di condizione con la relativa chiave complementare `aws:Ec2InstanceSourceVpc`. Utilizza questa chiave con `aws:Ec2InstanceSourceVpc` per assicurarti che la richiesta sia stata effettuata dallo stesso indirizzo IP privato a cui sono state consegnate le credenziali.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che il richiedente firma le richieste con una credenziale di ruolo Amazon EC2. Può essere utilizzata nelle policy IAM, nelle policy di controllo dei servizi, nelle policy degli endpoint VPC e nelle policy delle risorse.
- Tipo di dati: indirizzo IP
- Tipo di valore: valore singolo

> **Important**
>
> Questa chiave non deve essere utilizzata da sola in un'istruzione `Allow`. Per definizione, gli indirizzi IP privati non sono univoci a livello globale. Dovresti usare la chiave `aws:Ec2InstanceSourceVpc` ogni volta che utilizzi la chiave `aws:Ec2InstanceSourcePrivateIPv4` per specificare il VPC dal quale possono essere utilizzate le credenziali dell'istanza Amazon EC2.

> **Note**
>
> Questa chiave di condizione non è disponibile in EC2-Classic.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action":  "*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:Ec2InstanceSourceVpc": "${aws:SourceVpc}"
                },
                "Null": {
                    "ec2:SourceInstanceARN": "false"
                },
                "BoolIfExists": {
                    "aws:ViaAWSService": "false"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action":  "*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:Ec2InstanceSourcePrivateIPv4": "${aws:VpcSourceIp}"
                },
                "Null": {
                    "ec2:SourceInstanceARN": "false"
                },
                "BoolIfExists": {
                    "aws:ViaAWSService": "false"
                }
            }
        }
    ]
}
```

leggi: SourceIdentity

Utilizza questa chiave per confrontare l'identità di origine impostata dal principale con l'identità di origine specificata nella policy.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta dopo che è stata impostata un'identità di origine quando si assume un ruolo utilizzando qualsiasi comando CLI AWS STS assume-role o operazione API. AWS STS `AssumeRole`
- Tipo di dati: stringa
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave in una policy per consentire l'esecuzione di azioni AWS da parte dei responsabili che hanno impostato un'identità di origine quando assumono un ruolo. L'attività per l'identità di origine specificata del ruolo viene visualizzata in AWS CloudTrail. In questo modo è più facile per gli amministratori determinare chi o cosa ha eseguito le azioni con un ruolo. AWS

A differenza di `sts:RoleSessionName`, dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nel contesto della richiesta di tutte le operazioni intraprese dal ruolo. Il valore persiste nelle sessioni di ruolo successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

La `sts:SourceIdentity`chiave è presente nella richiesta quando il principale imposta inizialmente un'identità di origine assumendo un ruolo utilizzando qualsiasi comando CLI o operazione API di AWS STS assume-role. AWS STS `AssumeRole` La chiave `aws:SourceIdentity` è presente nella richiesta per tutte le operazioni eseguite con una sessione di ruolo con un set di identità di origine.

La policy di attendibilità del ruolo riportata di seguito per `CriticalRole` nell'account `111122223333` contiene una condizione per `aws:SourceIdentity`che impedisce a un principale senza un'identità di origine impostata su Saanvi o Diego di assumere il ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeRoleIfSourceIdentity",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::123456789012:role/CriticalRole"},
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringLike": {
                    "aws:SourceIdentity": ["Saanvi","Diego"]
```

```
                }
            }
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

ec2: RoleDelivery

Usa questa chiave per confrontare la versione del servizio di metadati dell'istanza nella richiesta firmata con le credenziali del ruolo IAM per Amazon EC2. Il servizio di metadati dell'istanza esegue la distinzione tra richieste IMDSv1 e IMDSv2 a seconda che, per una determinata richiesta, le intestazioni `PUT` o `GET`, che sono univoche per IMDSv2, siano presenti in tale richiesta.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che la sessione di ruolo viene creata da un'istanza Amazon EC2.
- Tipo di dati: numerico
- Tipo di valore: valore singolo
- Valori di esempio: 1,0, 2,0

Puoi configurare il servizio di metadati dell'istanza (IMDS) su ogni istanza in modo che il codice locale o gli utenti utilizzino IMDSv2. Quando specifichi l'utilizzo di IMDSv2, IMDSv1 non funziona più.

- Instance Metadata Service versione 1 (IMDSv1): un metodo di richiesta/risposta
- Servizio di metadati dell'istanza Versione 2 (IMDSv2): un metodo orientato alla sessione

Per informazioni su come configurare l'istanza per l'utilizzo di IMDSv2, consulta Configurare le opzioni dei metadati dell'istanza.

Nell'esempio seguente, l'accesso è negato se il RoleDelivery valore ec2: nel contesto della richiesta è 1.0 (IMDSv1). Questa dichiarazione politica può essere applicata in generale perché, se la richiesta non è firmata dalle credenziali del ruolo di Amazon EC2, non ha alcun effetto.

```
{
    "Version": "2012-10-17",
    "Statement": [
                {
```

```
            "Sid": "RequireAllEc2RolesToUseV2",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:RoleDelivery": "2.0"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni, consulta Politiche di esempio per l'utilizzo dei metadati delle istanze.

ec2: Arn SourceInstance

Utilizzate questa chiave per confrontare l'ARN dell'istanza da cui è stata generata la sessione del ruolo.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che la sessione di ruolo viene creata da un'istanza Amazon EC2.
- Tipo di dati: ARN
- Tipo di valore: valore singolo
- Valore di esempio: arn:aws:ec2:us-west- 2:1111:instance/instance-id

Per esempi di policy, consulta Consentire a un'istanza specifica di visualizzare le risorse in altri servizi. AWS

colla: RoleAssumed Di

Il AWS Glue servizio imposta questa chiave di condizione per ogni richiesta AWS API, in cui AWS Glue effettua una richiesta utilizzando un ruolo di servizio per conto del cliente (non tramite un endpoint di lavoro o sviluppatore, ma direttamente dal AWS Glue servizio). Utilizzate questa chiave per verificare se una chiamata a una AWS risorsa proviene dal AWS Glue servizio.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta quando si AWS Glue effettua una richiesta utilizzando un ruolo di servizio per conto del cliente.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

- Valore di esempio: questa chiave è sempre impostata su`glue.amazonaws.com`.

L'esempio seguente aggiunge una condizione per consentire al AWS Glue servizio di ottenere un oggetto da un bucket Amazon S3.

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::confidential-bucket/*",
    "Condition": {
        "StringEquals": {
            "glue:RoleAssumedBy": "glue.amazonaws.com"
        }
    }
}
```

glue: Servizio CredentialIssuing

Il AWS Glue servizio imposta questa chiave per ogni richiesta AWS API utilizzando un ruolo di servizio che proviene da un endpoint di lavoro o di sviluppo. Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene da un AWS Glue job o da un endpoint di sviluppo.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta quando si AWS Glue effettua una richiesta proveniente da un job o da un endpoint di sviluppo.
- Tipo di dati: stringa
- Tipo di valore: valore singolo
- Valore di esempio: questa chiave è sempre impostata su`glue.amazonaws.com`.

L'esempio seguente aggiunge una condizione associata a un ruolo IAM utilizzato da un AWS Glue job. Ciò garantisce che determinate azioni siano consentite/negate a seconda che la sessione di ruolo venga utilizzata per un ambiente di esecuzione del AWS Glue lavoro.

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::confidential-bucket/*",
    "Condition": {
        "StringEquals": {
            "glue:CredentialIssuingService": "glue.amazonaws.com"
```

```
        }
    }
}
```

lambda: Arn SourceFunction

Usa questa chiave per identificare l'ARN della funzione Lambda a cui sono state consegnate le credenziali del ruolo IAM. Il servizio Lambda imposta questa chiave per ogni richiesta AWS API proveniente dall'ambiente di esecuzione della funzione. Usa questa chiave per verificare se una chiamata a una AWS risorsa proviene dal codice di una funzione Lambda specifica. Lambda imposta questa chiave anche per alcune richieste che provengono dall'esterno dell'ambiente di esecuzione, come la scrittura di log CloudWatch e l'invio di tracce a X-Ray.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che viene richiamato il codice della funzione Lambda.
- Tipo di dati: ARN
- Tipo di valore: valore singolo
- Valore di esempio: arn:aws:lambda:us-east- 1:123456789012:function: TestFunction

L'esempio seguente consente a una funzione Lambda specifica di `s3:PutObject` accedere al bucket specificato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExampleSourceFunctionArn",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "ArnEquals": {
                    "lambda:SourceFunctionArn": "arn:aws:lambda:us-
east-1:123456789012:function:source_lambda"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni, consulta Lavorare con le credenziali dell'ambiente di esecuzione Lambda nella Guida per gli AWS Lambda sviluppatori.

ssm: Arn SourceInstance

Utilizza questa chiave per identificare l'ARN dell'istanza AWS Systems Manager gestita a cui sono state consegnate le credenziali del ruolo IAM. Questa chiave di condizione non è presente quando la richiesta proviene da un'istanza gestita con un ruolo IAM associato a un profilo di istanza Amazon EC2.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta ogni volta che le credenziali del ruolo vengono consegnate a un'istanza AWS Systems Manager gestita.
- Tipo di dati: ARN
- Tipo di valore: valore singolo
- Valore di esempio: arn:aws:ec2:us-west- 2:1111:instance/instance-id

archivio di identità: UserId

Utilizza questa chiave per confrontare l'identità della forza lavoro di IAM Identity Center nella richiesta firmata con l'identità specificata nella policy.

- Disponibilità: questa chiave viene inclusa quando il chiamante della richiesta è un utente di IAM Identity Center.
- Tipo di dati: stringa
- Tipo di valore: valore singolo
- Valore di esempio: 94482488-3041-7026-18f3-be45837cd0e4

Puoi trovare il nome di un utente in IAM Identity Center effettuando una richiesta all'API UserId Id utilizzando l'API, o l'SDK. GetUser AWS CLI AWS AWS

Proprietà della rete

Utilizzate le seguenti chiavi di condizione per confrontare i dettagli sulla rete da cui la richiesta ha avuto origine o da cui è passata la richiesta con le proprietà di rete specificate nella politica.

Indice

- leggi: SourceIp

- come: SourceVpc
- leggi: SourceVpce
- aws: VpcSource Ip

leggi: SourceIp

Utilizzare questa chiave per confrontare l'indirizzo IP del richiedente con l'indirizzo IP specificato nella policy. La chiave di condizione `aws:SourceIp` può essere utilizzata solo per intervalli di indirizzi IP pubblici.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta, tranne quando il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: indirizzo IP
- Tipo di valore: valore singolo

La chiave di condizione `aws:SourceIp` può essere utilizzata in una policy per consentire ai principali di effettuare richieste solo all'interno di un intervallo IP specificato.

> **Note**
>
> `aws:SourceIp` supporta indirizzi sia IPv4 sia IPv6 e intervalli di indirizzi IP. Per un elenco di quelli Servizi AWS che supportano IPv6, consulta Servizi AWS la pagina relativa al supporto IPv6 nella Amazon VPC User Guide.

Ad esempio, puoi collegare la seguente policy basata sull'identità a un ruolo IAM. Questa policy consente all'utente di collocare oggetti nel bucket di Amazon S3 `DOC-EXAMPLE-BUCKET3` se effettua la chiamata dall'intervallo di indirizzi IPv4 specificato. Questa politica consente inoltre a un AWS servizio che utilizza di eseguire questa operazione Inoltro delle sessioni di accesso per tuo conto.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PrincipalPutObjectIfIpAddress",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET3/*",
```

```
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "203.0.113.0/24"
                }
            }
        }
    ]
}
```

Se devi limitare l'accesso a reti che supportano l'indirizzamento sia IPv4 che IPv6, puoi includere l'indirizzo o gli intervalli di indirizzi IP IPv4 e IPv6 nella condizione della policy IAM. La seguente policy basata sull'identità consente all'utente di collocare oggetti nel bucket di Amazon S3 `DOC-EXAMPLE-BUCKET3` se l'utente effettua la chiamata dall'intervallo di indirizzi IPv4 o IPv6 specificato. Prima di includere gli intervalli di indirizzi IPv6 nella tua policy IAM, verifica che l'intervallo con Servizio AWS cui stai lavorando supporti IPv6. Per un elenco di quelli Servizi AWS che supportano IPv6, consulta Servizi AWS la pagina relativa al supporto IPv6 nella Amazon VPC User Guide.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PrincipalPutObjectIfIpAddress",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET3/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                }
            }
        }
    ]
}
```

Se la richiesta proviene da un host che utilizza un endpoint Amazon VPC, la chiave `aws:SourceIp` non è disponibile. Dovresti invece usare una chiave specifica per VPC come aws: Ip. VpcSource Per ulteriori informazioni sull'utilizzo degli endpoint VPC, consulta la sezione Gestione delle identità e degli accessi per endpoint VPC e servizi endpoint VPC nella Guida di AWS PrivateLink .

come: SourceVpc

Usa questa chiave per verificare se la richiesta viaggia attraverso il VPC a cui è collegato l'endpoint VPC. In una policy è possibile utilizzare questa chiave per consentire l'accesso solo a un VPC specifico. Per ulteriori informazioni, consulta Limitazione dell'accesso a un VPC specifico nella Guida per l'utente di Amazon Simple Storage Service.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

leggi: SourceVpce

Utilizzare questa chiave per confrontare l'identificatore dell'endpoint VPC della richiesta con l'ID endpoint specificato nella policy. In una policy è possibile utilizzare questa chiave per limitare l'accesso a un endpoint VPC specifico. Per ulteriori informazioni, consulta Limitazione dell'accesso a un endpoint VPC specifico nella Guida per l'utente di Amazon Simple Storage Service.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se il richiedente utilizza un endpoint VPC per effettuare la richiesta.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

aws: VpcSource Ip

Utilizzare questa chiave per confrontare l'indirizzo IP da cui è stata effettuata una richiesta con l'indirizzo IP specificato nella policy. In una policy, la chiave corrisponde solo se la richiesta proviene dall'indirizzo IP specificato e passa attraverso un endpoint VPC.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se la richiesta viene effettuata utilizzando un endpoint VPC.
- Tipo di dati: indirizzo IP
- Tipo di valore: valore singolo

Per ulteriori informazioni, consultare Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

> **Note**
>
> `aws:VpcSourceIp` supporta indirizzi sia IPv4 sia IPv6 e intervalli di indirizzi IP. Per un elenco di quelli Servizi AWS che supportano IPv6, consulta Servizi AWS la pagina relativa al supporto IPv6 nella Amazon VPC User Guide.

Le proprietà della risorsa

Utilizza le seguenti chiavi di condizione per confrontare i dettagli sulla risorsa oggetto della richiesta con le proprietà della risorsa specificate nella policy.

Indice

Leggi: ResourceAccount

Utilizza questa chiave per confrontare l'ID Account AWS del proprietario della risorsa richiesta con l'account della risorsa nella policy. Dopodiché, puoi consentire o negare l'accesso a tale risorsa in base all'account proprietario della stessa.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta per la maggior parte delle operazioni dei servizi. Le seguenti operazioni non supportano questa chiave:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`
  - Amazon Detective
    - `detective:AcceptInvitation`
  - Amazon Elastic Block Store: tutte le operazioni
  - Amazon EC2
    - `ec2:AcceptTransitGatewayPeeringAttachment`
    - `ec2:AcceptVpcEndpointConnections`
    - `ec2:AcceptVpcPeeringConnection`

    - `ec2:CopyImage`
    - `ec2:CopySnapshot`
    - `ec2:CreateTransitGatewayPeeringAttachment`
    - `ec2:CreateVolume`
    - `ec2:CreateVpcEndpoint`
    - `ec2:CreateVpcPeeringConnection`
    - `ec2:DeleteTransitGatewayPeeringAttachment`
    - `ec2:DeleteVpcPeeringConnection`
    - `ec2:RejectTransitGatewayPeeringAttachment`
    - `ec2:RejectVpcEndpointConnections`
    - `ec2:RejectVpcPeeringConnection`
- Amazon EventBridge
    - `events:PutEvents`— EventBridge `PutEvents` chiamate su un bus di eventi in un altro account, se tale bus di eventi è stato configurato come EventBridge destinazione tra più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere le autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
- Amazon GuardDuty
    - `guardduty:AcceptAdministratorInvitation`
- Amazon Macie
    - `macie2:AcceptInvitation`
- OpenSearch Servizio Amazon
    - `es:AcceptInboundConnection`
    - `es:CreateOutboundConnection`
- Amazon Route 53
    - `route53:AssociateVpcWithHostedZone`
    - `route53:CreateVPCAssociationAuthorization`
    - `route53:DeleteVPCAssociationAuthorization`
    - `route53:DisassociateVPCFromHostedZone`
    - `route53:ListHostedZonesByVPC`
- AWS Security Hub

    - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa
- Tipo di valore: valore singolo

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

Questa chiave è uguale all' Account AWS ID dell'account con le risorse valutate nella richiesta.

Per la maggior parte delle risorse presenti nel tuo account, l'ARN contiene l'ID account del proprietario della rispettiva risorsa. Per alcune risorse, come i bucket Amazon S3, l'ARN della risorsa non include l'ID account. I due esempi seguenti mostrano la differenza tra una risorsa il cui ARN contiene un ID account e un ARN Amazon S3 privo di un ID account:

- `arn:aws:iam::123456789012:role/AWSExampleRole`: ruolo IAM creato e di proprietà all'interno dell'account 123456789012.
- `arn:aws:s3:::DOC-EXAMPLE-BUCKET2`: bucket Amazon S3 creato e posseduto all'interno dell'account `111122223333`, non visualizzato nell'ARN.

Usa la AWS console, l'API o la CLI, per trovare tutte le tue risorse e gli ARN corrispondenti.

Scrivi una policy che nega le autorizzazioni alle risorse in base all'ID account del proprietario della risorsa. Ad esempio, la seguente policy basata sull'identità nega l'accesso alla risorsa specificata se la risorsa non appartiene all'account specificato.

Per utilizzare questa policy, sostituisci il testo segnaposto in corsivo con le tue informazioni.

> **Important**
>
> Questa policy non consente alcuna operazione. Utilizza invece l'effetto `Deny`, che nega esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyInteractionWithResourcesNotInSpecificAccount",
      "Action": "service:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:service:region:account:*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "account"
          ]
        }
      }
    }
  ]
}
```

Questa politica nega l'accesso a tutte le risorse per un AWS servizio specifico a meno che lo specificato non sia Account AWS proprietario della risorsa.

> **Note**
>
> Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceAccount` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi l'elemento `aws:ResourceAccount` nelle tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

aws: ResourceOrg Percorsi

Utilizzate questa chiave per confrontare il percorso AWS Organizations per la risorsa a cui si accede con il percorso nella policy. In una politica, questa chiave di condizione garantisce che la risorsa appartenga a un membro dell'account all'interno della radice o delle unità organizzative (OU) specificate in AWS Organizations. Un percorso AWS Organizations è una rappresentazione testuale della struttura di un'entità Organizations. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, consulta la sezione Informazioni sul percorso dell'entità AWS Organizations.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se l'account che possiede la risorsa è membro di un'organizzazione. Questa chiave della condizione globale non supporta le seguenti operazioni:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`
  - Amazon Detective
    - `detective:AcceptInvitation`
  - Amazon Elastic Block Store: tutte le operazioni
  - Amazon EC2
    - `ec2:AcceptTransitGatewayPeeringAttachment`
    - `ec2:AcceptVpcEndpointConnections`
    - `ec2:AcceptVpcPeeringConnection`
    - `ec2:CopyImage`
    - `ec2:CopySnapshot`
    - `ec2:CreateTransitGatewayPeeringAttachment`
    - `ec2:CreateVolume`
    - `ec2:CreateVpcEndpoint`
    - `ec2:CreateVpcPeeringConnection`
    - `ec2:DeleteTransitGatewayPeeringAttachment`
    - `ec2:DeleteVpcPeeringConnection`
    - `ec2:RejectTransitGatewayPeeringAttachment`
    - `ec2:RejectVpcEndpointConnections`
    - `ec2:RejectVpcPeeringConnection`

  - Amazon EventBridge

    - `events:PutEvents`— EventBridge `PutEvents` chiamate su un bus di eventi in un altro account, se tale bus di eventi è stato configurato come EventBridge destinazione tra più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere le autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
  - Amazon GuardDuty
    - `guardduty:AcceptAdministratorInvitation`
  - Amazon Macie
    - `macie2:AcceptInvitation`
  - OpenSearch Servizio Amazon
    - `es:AcceptInboundConnection`
    - `es:CreateOutboundConnection`
  - Amazon Route 53
    - `route53:AssociateVpcWithHostedZone`
    - `route53:CreateVPCAssociationAuthorization`
    - `route53:DeleteVPCAssociationAuthorization`
    - `route53:DisassociateVPCFromHostedZone`
    - `route53:ListHostedZonesByVPC`
  - AWS Security Hub
    - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

`aws:ResourceOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa

chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Chiavi di contesto multivalore.

Ad esempio, la seguente condizione restituisce `True` per le risorse appartenenti all'organizzazione `o-a1b2c3d4e5`. Quando si include un carattere jolly, è necessario utilizzare l'operatore di StringLikecondizione.

```
"Condition": {
      "ForAnyValue:StringLike": {
            "aws:ResourceOrgPaths":["o-a1b2c3d4e5/*"]
   }
}
```

La condizione seguente restituisce `True` alle risorse con l'ID dell'unità organizzativa `ou-ab12-11111111`. Assocerà le risorse di proprietà degli account collegati all'unità organizzativa ou-ab12-11111111 o a qualsiasi unità organizzativa figlia.

```
"Condition": { "ForAnyValue:StringLike" : {
     "aws:ResourceOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/*"]
}}
```

La seguente condizione restituisce `True` per le risorse di proprietà degli account collegati direttamente all'ID dell'unità organizzativa `ou-ab12-22222222`, ma non alle unità figlie. L'esempio seguente utilizza l'operatore StringEqualscondition per specificare il requisito di corrispondenza esatta per l'ID OU e non una corrispondenza con caratteri jolly.

```
"Condition": { "ForAnyValue:StringEquals" : {
     "aws:ResourceOrgPaths":["o-a1b2c3d4e5/r-ab12/ou-ab12-11111111/ou-ab12-22222222/"]
}}
```

> **Note**
>
> Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceOrgPaths` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi la chiave `aws:ResourceOrgPaths` nelle

tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio. Puoi creare una policy simile per limitare l'accesso alle risorse all'interno di un'unita organizzativa (OU) utilizzando la chiave `aws:ResourceOrgPaths`, tenendo conto delle risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

Leggi: ResourceOrg ID

Utilizza questa chiave per confrontare l'identificatore dell'organizzazione in AWS Organizations a cui appartiene la risorsa richiesta con l'identificatore specificato nella politica.

- Disponibilità: questa chiave viene inclusa nel contesto della richiesta solo se l'account che possiede la risorsa è membro di un'organizzazione. Questa chiave della condizione globale non supporta le seguenti operazioni:
  - AWS Audit Manager
    - `auditmanager:UpdateAssessmentFrameworkShare`
  - Amazon Detective
    - `detective:AcceptInvitation`
  - Amazon Elastic Block Store: tutte le operazioni
  - Amazon EC2
    - `ec2:AcceptTransitGatewayPeeringAttachment`
    - `ec2:AcceptVpcEndpointConnections`
    - `ec2:AcceptVpcPeeringConnection`
    - `ec2:CopyImage`
    - `ec2:CopySnapshot`
    - `ec2:CreateTransitGatewayPeeringAttachment`
    - `ec2:CreateVolume`
    - `ec2:CreateVpcEndpoint`
    - `ec2:CreateVpcPeeringConnection`
    - `ec2:DeleteTransitGatewayPeeringAttachment`
    - `ec2:DeleteVpcPeeringConnection`

    - `ec2:RejectTransitGatewayPeeringAttachment`
    - `ec2:RejectVpcEndpointConnections`
    - `ec2:RejectVpcPeeringConnection`
  - Amazon EventBridge
    - `events:PutEvents`— EventBridge `PutEvents` chiamate su un bus di eventi in un altro account, se tale bus di eventi è stato configurato come EventBridge destinazione tra più account prima del 2 marzo 2023. Per ulteriori informazioni, consulta Concedere le autorizzazioni per consentire eventi da altri AWS account nella Amazon EventBridge User Guide.
  - Amazon GuardDuty
    - `guardduty:AcceptAdministratorInvitation`
  - Amazon Macie
    - `macie2:AcceptInvitation`
  - OpenSearch Servizio Amazon
    - `es:AcceptInboundConnection`
    - `es:CreateOutboundConnection`
  - Amazon Route 53
    - `route53:AssociateVpcWithHostedZone`
    - `route53:CreateVPCAssociationAuthorization`
    - `route53:DeleteVPCAssociationAuthorization`
    - `route53:DisassociateVPCFromHostedZone`
    - `route53:ListHostedZonesByVPC`
  - AWS Security Hub
    - `securityhub:AcceptAdministratorInvitation`
- Tipo di dati: stringa
- Tipo di valore: valore singolo

> **Note**
>
> Per ulteriori considerazioni sulle operazioni non supportate di cui sopra, consulta il repository di esempi di policy del perimetro di dati.

Questa chiave globale restituisce l'ID dell'organizzazione della risorsa per una determinata richiesta. Consente di creare regole che si applicano a tutte le risorse di un'organizzazione che sono specificate nell'elemento `Resource` di una policy basata sull'identità. È possibile specificare l'ID organizzazione nell'elemento condizionale. Quando aggiungi e rimuovi degli account, le policy che includono la chiave `aws:ResourceOrgID` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

Ad esempio, la seguente policy impedisce al principale di aggiungere oggetti alla risorsa `policy-genius-dev`, a meno che la risorsa Amazon S3 non appartenga alla stessa organizzazione del principale che effettua la richiesta.

Important

Questa policy non consente alcuna operazione. Utilizza invece l'effetto `Deny`, che nega esplicitamente l'accesso a tutte le risorse elencate nell'istruzione che non appartengono all'account elencato. Utilizza questa policy in combinazione con altre policy che consentono l'accesso a risorse specifiche.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "DenyPutObjectToS3ResourcesOutsideMyOrganization",
    "Effect": "Deny",
    "Action": "s3:PutObject",
    "Resource": "arn:partition:s3:::policy-genius-dev/*",
    "Condition": {
      "StringNotEquals": {
        "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
      }
    }
  }
}
```

Note

Alcuni Servizi AWS richiedono l'accesso a risorse AWS di proprietà ospitate in un altro Account AWS. L'utilizzo di `aws:ResourceOrgID` nelle tue policy basate sull'identità potrebbe influire sulla capacità della tua identità di accedere a queste risorse.

Alcuni AWS servizi, ad esempio AWS Data Exchange, si basano sull'accesso a risorse esterne all'utente Account AWS per le normali operazioni. Se usi la chiave `aws:ResourceOrgID` nelle tue policy, includi istruzioni aggiuntive per creare delle esenzioni per tali servizi. La policy AWS: nega l'accesso alle risorse Amazon S3 al di fuori del tuo account tranne AWS Data Exchange di esempio illustra come negare l'accesso in base all'account della risorsa definendo al contempo delle eccezioni per le risorse di proprietà del servizio. Puoi creare una policy simile per limitare l'accesso alle risorse all'interno dell'organizzazione utilizzando la chiave `aws:ResourceOrgID`, tenendo conto delle risorse di proprietà del servizio.

Utilizza questo esempio di policy come modello per creare le tue policy personalizzate. Per ulteriori informazioni, consulta la documentazione del servizio.

Nel seguente video, scopri ulteriori informazioni su come utilizzare la chiave di condizione `aws:ResourceOrgID` in una policy.

Assicurati che le identità e le reti possano essere utilizzate solo per accedere a risorse attendibili.

aws:ResourceTag//tag-key

Utilizzare questa chiave per confrontare la coppia chiave-valore del tag specificata nella policy con la coppia chiave-valore associata alla risorsa. Ad esempio, puoi richiedere che l'accesso a una risorsa sia consentito solo se la risorsa dispone di una chiave di tag `"Dept"` collegata al valore `"Marketing"`. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse AWS.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando la risorsa richiesta dispone già di tag collegati o nelle richieste che creano una risorsa con un tag collegato. Questa chiave viene restituita solo per le risorse che supportano l'autorizzazione basata sui tag. È presente una chiave di contesto per ogni coppia chiave-valore del tag.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Questa chiave di contesto è formattata `"aws:ResourceTag/`*`tag-key`*`":"`*`tag-value`*`"` laddove *`tag-key`* e *`tag-value`* sono una coppia chiave e valore di tag. Per le chiavi e i valori dei tag non viene fatta la distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:ResourceTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della risorsa denominata `TagKey1` o `tagkey1`, ma non a entrambe.

Per esempi sull'utilizzo della chiave `aws:ResourceTag` per controllare l'accesso alle risorse IAM, consulta Controllo dell'accesso alle risorse AWS.

Per esempi di utilizzo della `aws:ResourceTag` chiave per controllare l'accesso ad altre AWS risorse, consultaControllo dell'accesso alle AWS risorse tramite tag.

Per un'esercitazione sull'utilizzo della chiave di condizione `aws:ResourceTag` per il controllo degli accessi basato su attributi (ABAC), consulta Tutorial IAM: definisci le autorizzazioni per accedere alle AWS risorse in base ai tag.

Proprietà della richiesta

Utilizzate le seguenti chiavi di condizione per confrontare i dettagli sulla richiesta stessa e il contenuto della richiesta con le proprietà della richiesta specificate nella politica.

Indice

Leggi: CalledVia

Utilizza questa chiave per confrontare i servizi nella policy con i servizi che hanno effettuato richieste per conto del principale IAM (utente o ruolo). Quando un principale effettua una richiesta a un AWS servizio, quel servizio potrebbe utilizzare le credenziali del principale per effettuare richieste successive ad altri servizi. La chiave `aws:CalledVia` contiene un elenco ordinato di ciascun servizio nella catena che ha effettuato le richieste per conto dell'entità principale.

Ad esempio, puoi usarlo AWS CloudFormation per leggere e scrivere da una tabella Amazon DynamoDB. DynamoDB utilizza quindi la crittografia fornita AWS Key Management Service da ().AWS KMS

- Disponibilità: questa chiave è presente nella richiesta quando un servizio che supporta `aws:CalledVia` utilizza le credenziali di un principale IAM per effettuare una richiesta a un altro servizio. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: String (elenco)
- Tipo di valore: multivalore

Per utilizzare la chiave di `aws:CalledVia` condizione in una politica, è necessario fornire i responsabili del servizio per consentire o rifiutare le richieste di AWS servizio. AWS supporta l'utilizzo dei seguenti principali di servizio con. `aws:CalledVia`

Principale del servizio
aoss.amazonaws.com
athena.amazonaws.com
backup.amazonaws.com
cloud9.amazonaws.com
cloudformation.amazonaws.com
databrew.amazonaws.com

Principale del servizio
dataexchange.amazonaws.com
dynamodb.amazonaws.com
imagebuilder.amazonaws.com
kms.amazonaws.com
mgn.amazonaws.com
nimble.amazonaws.com
omics.amazonaws.com
ram.amazonaws.com
robomaker.amazonaws.com
servicecatalog-appregistry.amazonaws.com
sqlworkbench.amazonaws.com
ssm-guiconnect.amazonaws.com

Per consentire o negare l'accesso quando qualsiasi servizio effettua una richiesta utilizzando le credenziali del principale, utilizzare la chiave di condizione `AWS: via AWSService`. Questa chiave di condizione supporta i AWS servizi.

La chiave `aws:CalledVia` è una chiave multivalore. Tuttavia, non è possibile applicare l'ordine utilizzando questa chiave in una condizione. Usando l'esempio precedente, l'utente 1 effettua una richiesta a AWS CloudFormation, che chiama DynamoDB, che a sua volta chiama AWS KMS. Si tratta di tre richieste distinte. L'ultima chiamata a AWS KMS viene eseguita dall'utente 1 tramite AWS CloudFormation e poi DynamoDB.

In questo caso, la chiave `aws:CalledVia` nel contesto della richiesta include `cloudformation.amazonaws.com` e `dynamodb.amazonaws.com`, in tale ordine. Se sei interessato al solo fatto che la chiamata sia stata effettuata tramite DynamoDB in qualche punto nella catena di richieste, puoi utilizzare questa chiave di condizione nella policy.

Ad esempio, la seguente politica consente di gestire la AWS KMS chiave denominata`my-example-key`, ma solo se DynamoDB è uno dei servizi richiedenti. L'operatore di condizione `ForAnyValue:StringEquals` assicura che DynamoDB sia uno dei servizi a effettuare chiamate. Se il principale effettua la chiamata AWS KMS direttamente, la condizione restituisce `false` e la richiesta non è consentita da questa policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KmsActionsIfCalledViaDynamodb",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:region:111122223333:key/my-example-key",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["dynamodb.amazonaws.com"]
                }
            }
        }
    ]
}
```

Se si desidera stabilire quale servizio effettua la prima o l'ultima chiamata nella catena, è possibile utilizzare le chiavi `aws:CalledViaFirst` e `aws:CalledViaLast`. Ad esempio, la seguente politica consente di gestire la chiave denominata in. `my-example-key` AWS KMS Queste AWS KMS operazioni sono consentite solo se nella catena sono state incluse più richieste. La prima richiesta deve essere fatta via AWS CloudFormation e l'ultima via DynamoDB. Se altri servizi fanno richieste nel mezzo della catena, l'operazione è ancora consentita.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KmsActionsIfCalledViaChain",
```

```
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:region:111122223333:key/my-example-key",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "cloudformation.amazonaws.com",
                    "aws:CalledViaLast": "dynamodb.amazonaws.com"
                }
            }
        }
    ]
}
```

Le chiavi `aws:CalledViaFirst` e `aws:CalledViaLast` sono presenti nella richiesta quando un servizio utilizza le credenziali di un'entità IAM per chiamare un altro servizio. Indicano il primo e l'ultimo servizio che ha effettuato chiamate nella catena di richieste. Ad esempio, supponiamo che AWS CloudFormation chiami un altro servizio denominato`X Service`, che chiama DynamoDB, che poi chiama. AWS KMS L'ultima chiamata a AWS KMS viene eseguita da `User 1` via AWS CloudFormation`X Service`, then e quindi da DynamoDB. È stato chiamato inizialmente tramite AWS CloudFormation e l'ultimo chiamato tramite DynamoDB.

aws: CalledVia Primo

Utilizza questa chiave per confrontare i servizi nella policy con il primo servizio che ha effettuato una richiesta per conto del principale IAM (utente o ruolo). Per ulteriori informazioni, consulta `aws:CalledVia`.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio utilizza le credenziali di un principale IAM per effettuare almeno un'altra richiesta a un servizio diverso. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

aws: CalledVia Ultimo

Utilizza questa chiave per confrontare i servizi nella policy con l'ultimo servizio che ha effettuato una richiesta per conto del principale IAM (utente o ruolo). Per ulteriori informazioni, consulta `aws:CalledVia`.

- Disponibilità: questa chiave è presente nella richiesta quando un servizio utilizza le credenziali di un principale IAM per effettuare almeno un'altra richiesta a un servizio diverso. Questa chiave non è presente se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. Questa chiave non è presente anche quando il principale effettua la chiamata direttamente.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

AWS: via AWSService

Usa questa chiave per verificare se un AWS servizio effettua una richiesta a un altro servizio per tuo conto.

La chiave di contesto della richiesta restituisce `true` quando un servizio utilizza le credenziali di un'entità IAM per effettuare una richiesta per conto del principale. La chiave di contesto restituisce `false` se il servizio utilizza un ruolo di servizio oppure un ruolo collegato ai servizi per effettuare una chiamata per conto del principale. La chiave di contesto della richiesta restituisce anche `false` quando il principale effettua direttamente la chiamata.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: booleano
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per consentire o negare l'accesso in base al fatto che una richiesta sia stata effettuata da un servizio.

leggi: CurrentTime

Utilizzare questa chiave per confrontare la data e l'ora della richiesta con la data e l'ora specificate nella policy. Per visualizzare una policy di esempio che utilizza la chiave di condizione, consulta AWS: consente l'accesso in base alla data e all'ora.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: data
- Tipo di valore: valore singolo

leggi: EpochTime

Utilizzare questa chiave per confrontare la data e l'ora della richiesta in formato epoch o ora Unix con il valore specificato nella policy. Questa chiave accetta anche il numero di secondi dal 1 gennaio 1970.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: data, numerico
- Tipo di valore: valore singolo

aws:Referer

Utilizzare questa chiave per confrontare il referrer della richiesta nel browser client con il referrer specificato nella policy. Il valore del contesto della richiesta `aws:referer` viene fornito dal chiamante in un'intestazione HTTP. L'intestazione `Referer` viene inclusa in una richiesta del browser Web quando si seleziona un link in una pagina Web. L'intestazione `Referer` contiene l'URL della pagina Web in cui è stato selezionato il link.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo se la richiesta alla AWS risorsa è stata richiamata mediante un collegamento dall'URL di una pagina Web nel browser. Questa chiave non è inclusa per le richieste a livello di programmazione perché non utilizza un link del browser per accedere alla risorsa AWS .
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Ad esempio, è possibile accedere a un oggetto Amazon S3 direttamente utilizzando un URL o utilizzando l'invocazione API diretta. Per ulteriori informazioni, consulta Operazioni delle API Amazon S3 direttamente tramite un browser Web. Quando si accede a un oggetto Amazon S3 da un URL presente in una pagina Web, l'URL della pagina Web di origine viene utilizzato in `aws:referer`. Quando si accede a un oggetto Amazon S3 digitando l'URL nel browser, `aws:referer` non è presente. Quando si richiama direttamente l'API, `aws:referer` non è presente. È possibile utilizzare

la chiave di condizione `aws:referer` in una policy per autorizzare le richieste effettuate da un referente specifico, ad esempio un link su una pagina Web nel dominio dell'azienda.

> **Warning**
>
> Questa chiave deve essere utilizzata con attenzione. È pericoloso includere un valore dell'intestazione del referrer pubblicamente noto. Parti non autorizzate possono utilizzare browser modificati o personalizzati per fornire qualsiasi valore `aws:referer` scelto. Di conseguenza, non `aws:referer` deve essere utilizzato per impedire a parti non autorizzate di effettuare AWS richieste dirette. È disponibile solo per consentire ai clienti di proteggere i propri contenuti digitali, come i contenuti memorizzati su Amazon S3, da riferimenti su siti di terze parti non autorizzate.

Leggi: RequestedRegion

Utilizza questa chiave per confrontare la AWS regione chiamata nella richiesta con la regione specificata nella politica. È possibile utilizzare questa chiave di condizione globale per controllare quali regioni possono essere richieste. Per visualizzare le AWS regioni per ogni servizio, consulta Endpoint e quote del servizio in. Riferimenti generali di Amazon Web Services

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

I servizi globali, ad esempio IAM, dispongono di un unico endpoint. Poiché questo endpoint si trova fisicamente nella regione Stati Uniti orientali (Virginia settentrionale), le chiamate IAM vengono sempre effettuate alla regione us-east-1. Ad esempio, se crei una policy che nega l'accesso a tutti i servizi qualora la regione richiesta non fosse us-west-2 allora le chiamate IAM non andranno mai a buon fine. Per vedere un esempio di come ovviare a questo problema, vedi NotAction con Deny.

> **Note**
>
> La chiave di condizione `aws:RequestedRegion` consente di controllare quale endpoint di un servizio è richiamato ma non controlla l'impatto dell'operazione. Alcuni servizi hanno impatti su più regioni.
> Ad esempio, Amazon S3 dispone di operazioni API che si estendono a diverse regioni.

- È possibile richiamare `s3:PutBucketReplication` in una regione (che è interessata dalla chiave di condizione `aws:RequestedRegion`) e altre regioni vengono interessate in base alle impostazioni di configurazione delle repliche.
- Puoi richiamare `s3:CreateBucket`per creare un bucket in un'altra regione e utilizzare la `s3:LocationConstraint`chiave di condizione per controllare le regioni applicabili.

È possibile utilizzare questa chiave di contesto per limitare l'accesso ai AWS servizi all'interno di un determinato insieme di regioni. Ad esempio, la seguente policy consente a un utente di visualizzare tutte le istanze Amazon EC2 nella AWS Management Console. Tuttavia consente loro di modificare solo le istanze in Irlanda (eu-west-1), a Londra (eu-west-2) o Parigi (eu-west-3).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "InstanceConsoleReadOnly",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:Export*",
                "ec2:Get*",
                "ec2:Search*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "InstanceWriteRegionRestricted",
            "Effect": "Allow",
            "Action": [
                "ec2:Associate*",
                "ec2:Import*",
                "ec2:Modify*",
                "ec2:Monitor*",
                "ec2:Reset*",
                "ec2:Run*",
                "ec2:Start*",
                "ec2:Stop*",
                "ec2:Terminate*"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": [
                        "eu-west-1",
                        "eu-west-2",
                        "eu-west-3"
                    ]
                }
            }
        }
    ]
}
```

aws:RequestTag//tag-key

Utilizzare questa chiave per confrontare la coppia chiave-valore del tag passata nella richiesta con la coppia del tag specificata nella policy. Ad esempio, è possibile controllare che la richiesta includa la chiave del tag `"Dept"` e che abbia il valore `"Accounting"`. Per ulteriori informazioni, consulta Controllo dell'accesso durante le richieste AWS.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta quando le coppie chiave-valore dei tag vengono passate nella richiesta. Quando più tag vengono passati nella richiesta, è presente una chiave di contesto per ogni coppia chiave-valore dei tag.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Questa chiave di contesto è formattata `"aws:RequestTag/`*`tag-key`*`":"`*`tag-value`*`"` laddove *`tag-key`* e *`tag-value`* sono una coppia chiave e valore di tag. Per le chiavi e i valori dei tag non viene fatta la distinzione tra maiuscole e minuscole. Questo significa che se specifichi `"aws:RequestTag/TagKey1": "Value1"` nell'elemento condizione della policy, la condizione corrisponderà a una chiave di tag della richiesta denominata `TagKey1` o `tagkey1`, ma non a entrambi.

Questo esempio mostra che, sebbene la chiave abbia un singolo valore, è comunque possibile utilizzare più coppie chiave-valore in una richiesta se le chiavi sono diverse.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
```

```
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:::instance/*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/environment": [
          "preprod",
          "production"
        ],
        "aws:RequestTag/team": [
          "engineering"
        ]
      }
    }
  }
}
```

leggi: TagKeys

Utilizzare questa chiave per confrontare le chiavi dei tag in una richiesta con quelle specificate nella policy. Nell'utilizzo delle policy per controllare gli accessi tramite tag, è consigliabile utilizzare la chiave di condizione `aws:TagKeys` per definire le chiavi di tag ammesse. Per esempi di policy e ulteriori informazioni, consultare the section called “Controllo dell'accesso in base alle chiavi di tag”

- Disponibilità: questa chiave è inclusa nel contesto della richiesta se l'operazione supporta il passaggio di tag nella richiesta.
- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

Questa chiave di contesto ha il formato `"aws:TagKeys":"`*`tag-key`*`"`, dove *`tag-key`* è una lista di chiavi di tag senza valori (ad esempio `["Dept","Cost-Center"]`).

Poiché è possibile includere più coppie chiave-valore dei tag in una richiesta, il contenuto della richiesta potrebbe essere una richiesta multivalore. In questo caso, devi usare gli operatori su set `ForAllValues` o `ForAnyValue`. Per ulteriori informazioni, consulta Chiavi di contesto multivalore.

Alcuni servizi supportano il tagging con operazioni sulle risorse, come la creazione, la modifica o l'eliminazione di una risorsa. Per consentire il tagging e le operazioni come chiamata singola, è necessario creare una policy che comprende le operazioni di tagging e di modifica della risorsa. È quindi possibile utilizzare la chiave di condizione `aws:TagKeys` per implementare nella richiesta specifiche chiavi di tag. Ad esempio, per limitare i tag quando un utente crea uno snapshot Amazon

EC2, è necessario includere nella policy l'operazione di creazione `ec2:CreateSnapshot` e l'operazione di tagging `ec2:CreateTags`. Per visualizzare una policy per questo scenario che utilizza`aws:TagKeys`, consulta Creating a Snapshot with Tags nella Amazon EC2 User Guide.

leggi: SecureTransport

Utilizzare questa chiave per verificare se la richiesta è stata inviata utilizzando SSL. Il contesto della richiesta restituisce `true` o `false`. In una policy, è possibile consentire operazioni specifiche solo se la richiesta viene inviata tramite SSL.

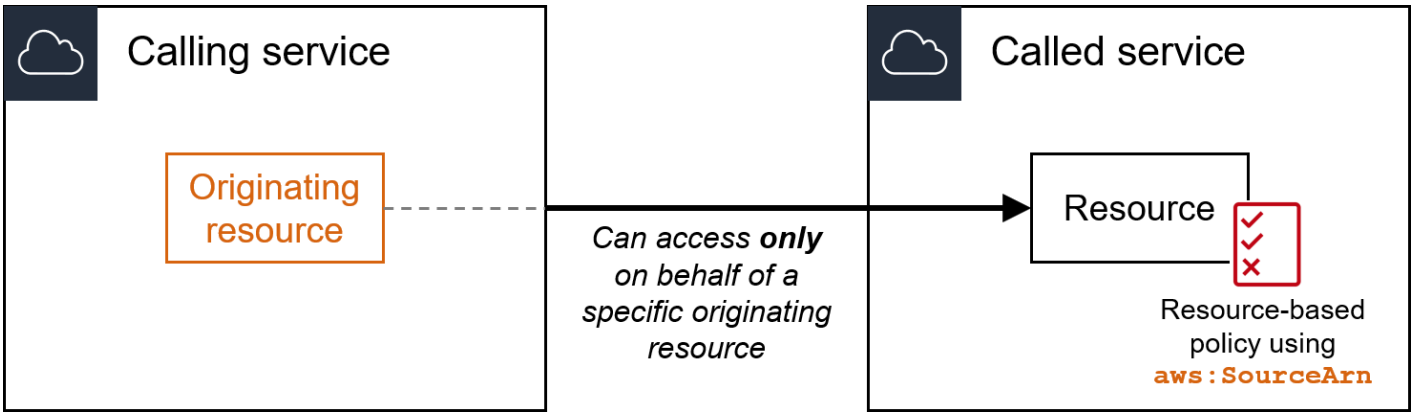
- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: booleano
- Tipo di valore: valore singolo

leggi: SourceArn

Usa questa chiave per confrontare l'Amazon Resource Name (ARN) della risorsa che effettua una service-to-service richiesta con l'ARN specificato nella policy, ma solo quando la richiesta viene effettuata da un responsabile del servizio. AWS Quando l'ARN d'origine include l'ID account, non è necessario utilizzare `aws:SourceAccount` con `aws:SourceArn`.

Questa chiave non funziona con l'ARN del principale che effettua la richiesta. Utilizza invece `Leggi: PrincipalArn`.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un responsabile del AWS servizio per conto di una risorsa per la quale la configurazione ha attivato la richiesta. service-to-service Il servizio chiamante passa l'ARN della risorsa originale al servizio chiamato.



Le seguenti integrazioni di servizio non supportano questa chiave di condizione globale:

Servizio di chiamata (principale del servizio)	Servizio chiamato (policy basata sulle risorse)	Descrizione
logdelivery.elb.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
logdelivery.elasticloadbalancing.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3

> **Note**
>
> Non tutte le integrazioni di servizi con AWS Security Token Service (AWS STS) e AWS Key Management Service (AWS KMS) sono supportate. Per ulteriori informazioni, consulta la documentazione del servizio di chiamata. L'utilizzo delle politiche chiave incluse `aws:SourceArn` nel KMS per le chiavi utilizzate Servizi AWS tramite KMS key grants può causare un comportamento imprevisto.

- Tipo di dati: ARN, String

  AWS consiglia di utilizzare operatori ARN anziché operatori stringa quando si confrontano gli ARN.

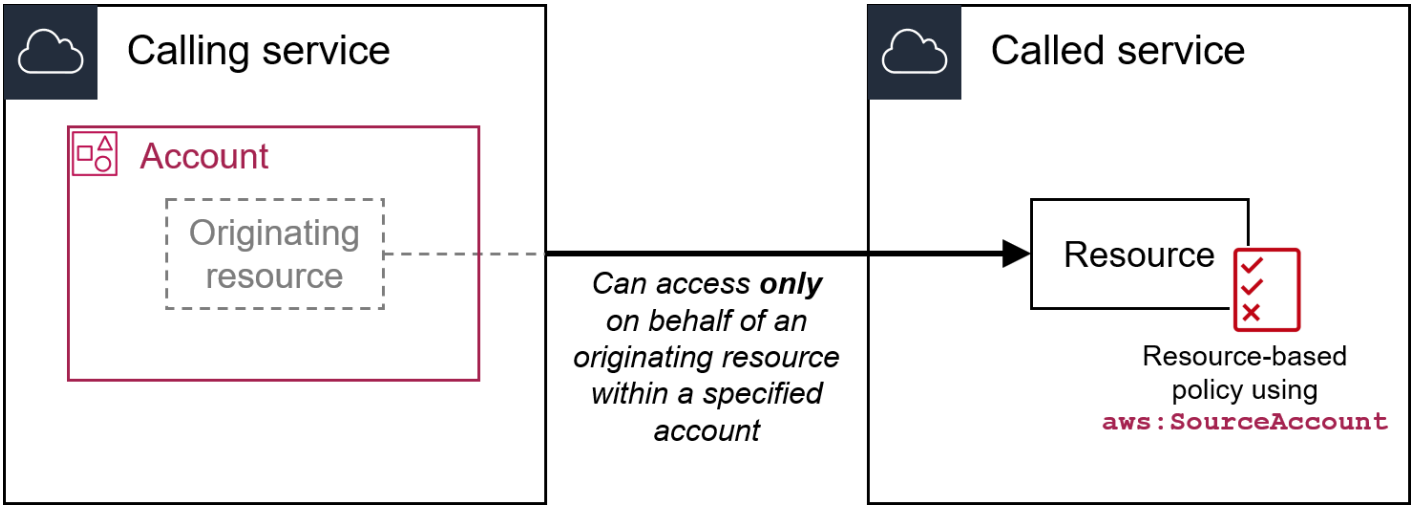
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per evitare che un AWS servizio venga utilizzato come sostituto confuso durante le transazioni tra servizi. Utilizzate questa chiave solo nelle politiche basate sulle risorse in cui il `Principal` è principale. Servizio AWS Imposta il valore di questa chiave di condizione sull'ARN della risorsa nella richiesta. Ad esempio, quando un aggiornamento di un bucket Amazon S3 attiva una pubblicazione sull'argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish`. Nella policy di argomento che consente l'operazione `sns:Publish`, imposta il valore della chiave di condizione sull'ARN del bucket Amazon S3. Per informazioni su come e quando è consigliata questa chiave di condizione, consulta la documentazione relativa AWS ai servizi che stai utilizzando.

leggi: SourceAccount

Utilizza questa chiave per confrontare l'ID account della risorsa che effettua una service-to-service richiesta con l'ID dell'account specificato nella politica, ma solo quando la richiesta viene effettuata da un responsabile del AWS servizio.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un responsabile del AWS servizio per conto di una risorsa per la quale la configurazione ha attivato la service-to-service richiesta. Il servizio chiamante deve passare l'ID account della risorsa originale al servizio chiamato.



Le seguenti integrazioni di servizio non supportano questa chiave di condizione globale:

Servizio di chiamata (principale del servizio)	Servizio chiamato (policy basata sulle risorse)	Descrizione
logdelivery.elb.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
logdelivery.elasticloadbalancing.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3

> **Note**
>
> Non tutte le integrazioni di servizi con AWS Security Token Service (AWS STS) e AWS Key Management Service (AWS KMS) sono supportate. Per ulteriori informazioni, consulta la documentazione del servizio di chiamata. L'utilizzo delle politiche chiave incluse `aws:SourceAccount` nel KMS per le chiavi utilizzate Servizi AWS tramite KMS key grants può causare un comportamento imprevisto.

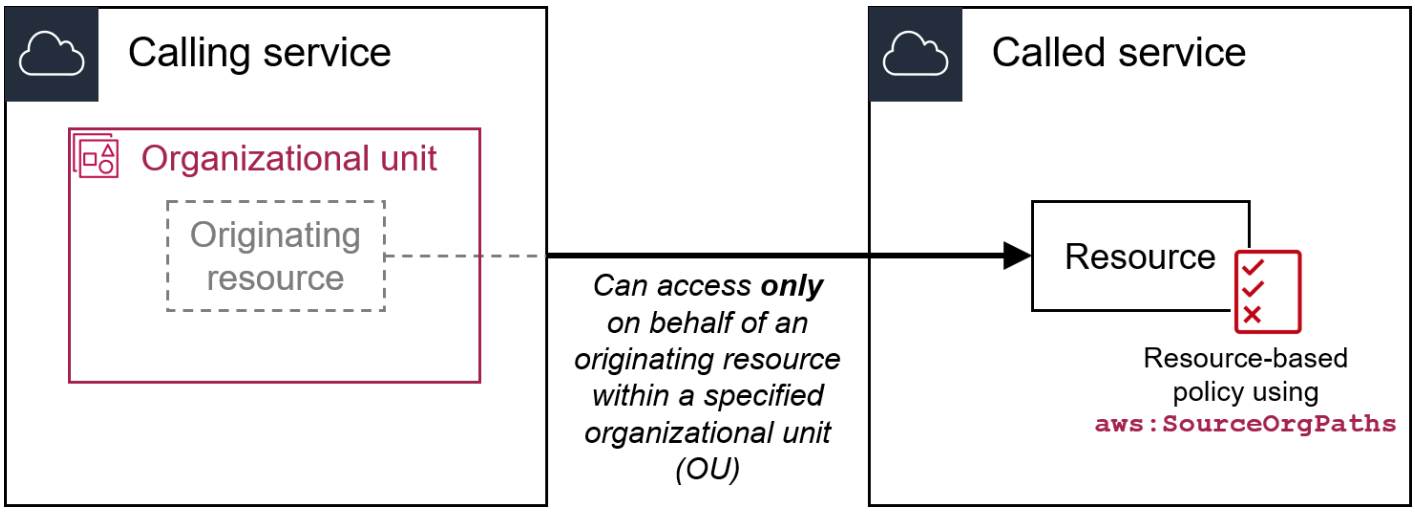
- Tipo di dati: stringa
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per evitare che un AWS servizio venga utilizzato come sostituto confuso durante le transazioni tra servizi. Utilizzate questa chiave solo nelle politiche basate sulle risorse in cui il `Principal` è principale. Servizio AWS Imposta il valore di questa chiave di condizione sull'ID account della risorsa nella richiesta. Ad esempio, quando un aggiornamento di un bucket Amazon S3 attiva una pubblicazione sull'argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish`. Nella policy di argomento che autorizza l'operazione `sns:Publish`, imposta il valore della chiave di condizione sull'ID account del bucket Amazon S3. Per informazioni su come e quando queste chiavi di condizione sono consigliate, consulta la documentazione relativa AWS ai servizi che stai utilizzando.

aws: SourceOrg Percorsi

Utilizzate questa chiave per confrontare il AWS Organizations percorso della risorsa che effettua una service-to-service richiesta con il percorso dell'organizzazione specificato nella politica, ma solo quando la richiesta viene effettuata da un responsabile del AWS servizio. Un percorso Organizations è una rappresentazione testuale della struttura di un'entità Organizations. Per ulteriori informazioni sull'utilizzo e la conoscenza dei percorsi, consulta Comprendere il percorso dell'entità AWS Organizations.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa di proprietà di un account membro di un'organizzazione. Il servizio chiamante passa il percorso dell'organizzazione della risorsa originale al servizio chiamato.



Le seguenti integrazioni di servizio non supportano questa chiave di condizione globale:

Servizio di chiamata (principale del servizio)	Servizio chiamato (policy basata sulle risorse)	Descrizione
logdelivery.elb.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
logdelivery.elasticloadbalancing.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
Tutti i principali di servizio	Bot di Amazon Lex	Consenti Servizi AWS l'uso del bot Amazon Lex

> **Note**
>
> Non tutte le integrazioni di servizi con AWS Security Token Service (AWS STS) e AWS Key Management Service (AWS KMS) sono supportate. Per ulteriori informazioni, consulta la documentazione del servizio di chiamata. L'utilizzo delle politiche chiave incluse

`aws:SourceOrgPaths` nel KMS per le chiavi utilizzate Servizi AWS tramite KMS key grants può causare un comportamento imprevisto.

- Tipo di dati: stringa (elenco)
- Tipo di valore: multivalore

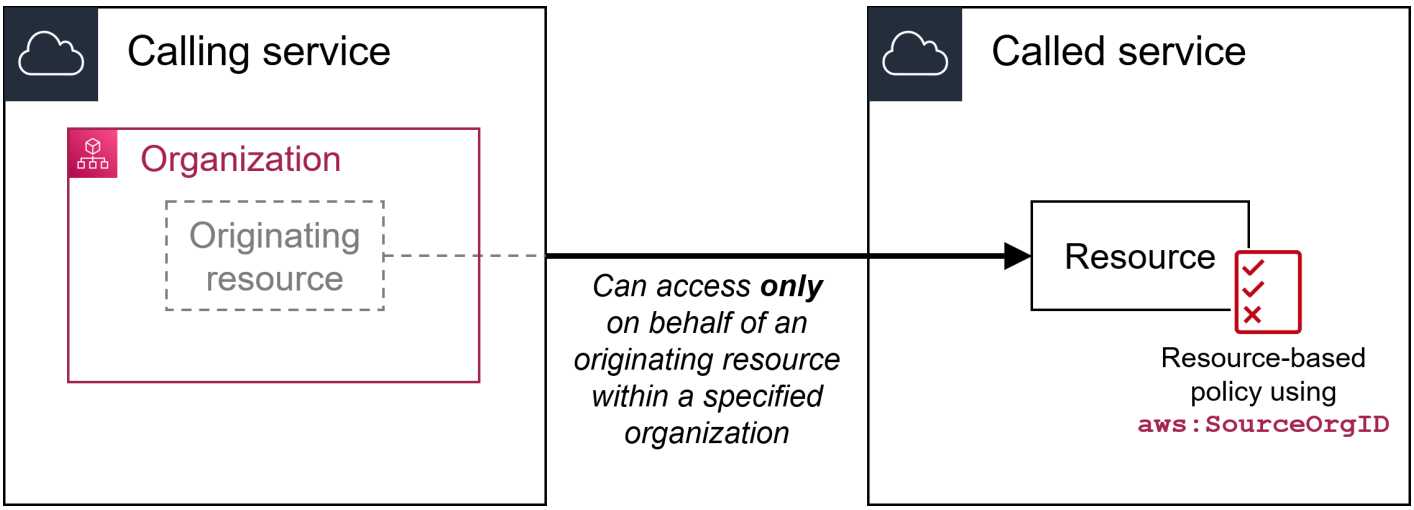
È possibile utilizzare questa chiave di condizione per evitare che un AWS servizio venga utilizzato come sostituto confuso durante le transazioni tra servizi. Utilizzate questa chiave solo nelle politiche basate sulle risorse in cui il `Principal` è principale. Servizio AWS Imposta il valore di questa chiave di condizione sul percorso dell'organizzazione della risorsa nella richiesta. Ad esempio, quando un aggiornamento di un bucket Amazon S3 attiva una pubblicazione sull'argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish`. Nella policy di argomento che consente l'operazione `sns:Publish`, imposta il valore della chiave di condizione sul percorso dell'organizzazione del bucket Amazon S3. Per informazioni su come e quando è consigliata questa chiave di condizione, consulta la documentazione relativa AWS ai servizi che stai utilizzando.

`aws:SourceOrgPaths` è una chiave di condizione multivalore. Le chiavi multivalore possono avere più di un valore nel contesto della richiesta. È necessario utilizzare gli operatori di insieme `ForAnyValue` o `ForAllValues` con gli operatori di condizione di stringa quando si utilizza questa chiave. Per ulteriori informazioni sulle chiavi di condizione multivalore, consultare Chiavi di contesto multivalore.

aws: SourceOrg ID

Utilizza questa chiave per confrontare l'ID dell'organizzazione della risorsa che effettua una service-to-service richiesta con l'ID dell'organizzazione specificato nella politica, ma solo quando la richiesta viene effettuata da un responsabile del AWS servizio. Quando aggiungi e rimuovi gli account da un'organizzazione in AWS Organizations, le policy che includono la chiave `aws:SourceOrgID` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

- Disponibilità: questa chiave è inclusa nel contesto della richiesta solo quando la chiamata alla risorsa viene effettuata direttamente da un principale del servizio AWS per conto di una risorsa di proprietà di un account membro di un'organizzazione. Il servizio chiamante passa l'ID dell'organizzazione della risorsa originale al servizio chiamato.



Le seguenti integrazioni di servizio non supportano questa chiave di condizione globale:

Servizio di chiamata (principale del servizio)	Servizio chiamato (policy basata sulle risorse)	Descrizione
logdelivery.elb.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
logdelivery.elasticloadbalancing.amazonaws.com	Bucket Amazon S3	Abilita la registrazione degli accessi con Elastic Load Balancing nel bucket Amazon S3
Tutti i principali di servizio	Bot di Amazon Lex	Consenti Servizi AWS l'uso del bot Amazon Lex

> **Note**
>
> Non tutte le integrazioni di servizi con AWS Security Token Service (AWS STS) e AWS Key Management Service (AWS KMS) sono supportate. Per ulteriori informazioni, consulta la documentazione del servizio di chiamata. L'utilizzo delle politiche chiave incluse

`aws:SourceOrgID` nel KMS per le chiavi utilizzate Servizi AWS tramite KMS key grants può causare un comportamento imprevisto.

- Tipo di dati: stringa
- Tipo di valore: valore singolo

È possibile utilizzare questa chiave di condizione per evitare che un AWS servizio venga utilizzato come sostituto confuso durante le transazioni tra servizi. Utilizzate questa chiave solo nelle politiche basate sulle risorse in cui il `Principal` è principale. Servizio AWS Imposta il valore di questa chiave di condizione sull'ID dell'organizzazione della risorsa nella richiesta. Ad esempio, quando un aggiornamento di un bucket Amazon S3 attiva una pubblicazione sull'argomento Amazon SNS, il servizio Amazon S3 richiama l'operazione API `sns:Publish`. Nella policy di argomento che consente l'operazione `sns:Publish`, imposta il valore della chiave di condizione sull'ID dell'organizzazione del bucket Amazon S3. Per informazioni su come e quando è consigliata questa chiave di condizione, consulta la documentazione relativa AWS ai servizi che stai utilizzando.

leggi: UserAgent

Utilizzare questa chiave per confrontare l'applicazione client del richiedente con l'applicazione specificata nella policy.

- Disponibilità: questa chiave è sempre inclusa nel contesto della richiesta.
- Tipo di dati: stringa
- Tipo di valore: valore singolo

Warning

Questa chiave deve essere utilizzata con attenzione. Poiché il valore `aws:UserAgent` viene fornito dall'intermediario in un'intestazione HTTP, le parti non autorizzate possono utilizzare browser modificati o personalizzati per fornire qualsiasi valore `aws:UserAgent` da essi scelto. Di conseguenza, non `aws:UserAgent` deve essere utilizzato per impedire a parti non autorizzate di effettuare AWS richieste dirette. Puoi utilizzarlo per consentire solo applicazioni client specifiche e solo dopo il test della policy.

Altre chiavi di condizione cross-service

AWS STS supporta chiavi di condizione di federazione basate su SAML e chiavi di condizione tra servizi per la federazione OIDC. Queste chiavi sono disponibili quando un utente federato tramite SAML esegue operazioni in altri servizi. AWS

chiavi contestuali IAM e AWS STS condition

Puoi utilizzare l'`Condition`elemento in una policy JSON per testare il valore delle chiavi incluse nel contesto di richiesta di tutte le AWS richieste. Queste chiavi forniscono informazioni sulla richiesta in sé o sulle risorse a cui la richiesta fa riferimento. È possibile controllare che le chiavi abbiano determinati valori prima di consentire l'operazione richiesta dall'utente. Ciò consente un controllo granulare sulla corrispondenza o meno delle istruzioni della policy JSON rispetto a una richiesta API in ingresso. Per informazioni su come utilizzare l'elemento `Condition` in una policy JSON, consulta Elementi delle policy JSON IAM: Condition.

Questo argomento descrive le chiavi definite e fornite dal servizio IAM (con un `iam:` prefisso) e dal servizio AWS Security Token Service (AWS STS) (con un `sts:` prefisso). Diversi altri AWS servizi forniscono anche chiavi specifiche del servizio che sono rilevanti per le azioni e le risorse definite da quel servizio. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per i AWS servizi. La documentazione relativa a un servizio che supporta le chiavi di condizione contiene spesso ulteriori informazioni. Ad esempio, per informazioni sulle chiavi che puoi utilizzare nelle policy per le risorse Amazon S3, consulta Chiavi di policy Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Argomenti

Chiavi disponibili per IAM

È possibile utilizzare le seguenti chiavi di condizione nelle policy che controllano l'accesso alle risorse IAM:

Sono: AssociatedResourceArn

Lavora con operatori ARN.

Specifica l'ARN della risorsa a cui verrà associato questo ruolo al servizio di destinazione. La risorsa in genere appartiene al servizio a cui l'entità sta passando il ruolo. A volte, la risorsa potrebbe appartenere a un terzo servizio. Ad esempio, potresti passare un ruolo a Amazon EC2 Auto Scaling che può essere utilizzato in un'istanza di Amazon EC2. In questo caso, la condizione corrisponderebbe all'ARN dell'istanza Amazon EC2.

Questa chiave di condizione si applica solo all'PassRoleazione in una politica. Non può essere usata per limitare altre operazioni.

Utilizzare questa chiave di condizione in un criterio per consentire a un'entità di passare un ruolo, ma solo se tale ruolo è associato alla risorsa specificata. È possibile utilizzare caratteri jolly (*) per consentire operazioni eseguite su un tipo specifico di risorsa senza limitare la regione o l'ID risorsa. Ad esempio, è possibile consentire a un utente o a un ruolo IAM di passare qualsiasi ruolo al servizio Amazon EC2 da utilizzare con istanze nella regione `us-east-1` o `us-west-1`. L'utente o il ruolo IAM non è autorizzato a passare ruoli ad altri servizi. Inoltre, non consente ad Amazon EC2 di utilizzare il ruolo con istanze in altre regioni.

```
{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {"iam:PassedToService": "ec2.amazonaws.com"},
        "ArnLike": {
            "iam:AssociatedResourceARN": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*",
                "arn:aws:ec2:us-west-1:111122223333:instance/*"
            ]
        }
    }
}
```

> **Note**
>
> AWS servizi che supportano iam: supportano PassedToService anche questa chiave di condizione.

sono: AWSServiceName

Lavora con operatori stringa.

Speciifica il AWS servizio a cui è associato questo ruolo.

In questo esempio, consenti a un'entità di creare un ruolo collegato ai servizi se il nome del servizio è access-analyzer.amazonaws.com.

```
{
   "Version": "2012-10-17",
   "Statement": [{
       "Effect": "Allow",
       "Action": "iam:CreateServiceLinkedRole",
       "Resource": "*",
       "Condition": {
         "StringLike": {
           "iam:AWSServiceName": "access-analyzer.amazonaws.com"
         }
       }
     }]
 }
```

iam:FIDO-certification

Lavora con operatori stringa.

Verifica il livello di certificazione FIDO del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L1plus, L2, L2plus, L3, L3plus

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione FIDO Level 1 plus per il tuo dispositivo.

```
{
      "Version": "2012-10-17",
      "Statement": [{
            "Effect": "Allow",
```

```
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-certification": "L1plus"
                }
            }
        }
    ]

}
```

iam:FIDO-FIPS-140-2-certification

Lavora con operatori stringa.

Verifica il livello di certificazione di convalida FIPS-140-2 del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L2, L3, L4

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione FIPS-140-2 Level 2 per il tuo dispositivo.

```
{
      "Version": "2012-10-17",
      "Statement": [{
            "Effect": "Allow",
```

```
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-2-certification": "L2"
                }
            }
        }
    ]

}
```

iam:FIDO-FIPS-140-3-certification

Lavora con operatori stringa.

Verifica il livello di certificazione di convalida FIPS-140-3 del dispositivo MFA al momento della registrazione di una chiave di sicurezza FIDO. La certificazione del dispositivo viene recuperata dal FIDO Alliance Metadata Service (MDS). Se lo stato o il livello di certificazione della chiave di sicurezza FIDO cambia, questa non verrà aggiornata a meno che la registrazione del dispositivo non sia stata annullata e poi effettuata nuovamente per recuperare le informazioni di certificazione aggiornate.

Valori possibili di L1, L2, L3, L4

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione FIPS-140-3 Level 3 per il tuo dispositivo.

```
{
      "Version": "2012-10-17",
      "Statement": [{
```

```
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L3"
                }
            }
        }
    ]

}
```

sono: RegisterSecurityKey

Lavora con operatori stringa.

Verifica lo stato corrente dell'abilitazione dei dispositivi MFA.

Valori possibili di `Create` o `Activate`.

In questo esempio, registri una chiave di sicurezza e recuperi la certificazione FIPS-140-3 Level 1 per il tuo dispositivo.

```
{
      "Version": "2012-10-17",
      "Statement": [{
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "iam:RegisterSecurityKey" : "Create"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:EnableMFADevice",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:RegisterSecurityKey" : "Activate",
                    "iam:FIDO-FIPS-140-3-certification": "L1"
                }
            }
        }
    ]

}
```

Io sono: OrganizationsPolicyId

Lavora con operatori stringa.

Verifica che la politica con l' AWS Organizations ID specificato corrisponda alla politica utilizzata nella richiesta. Per visualizzare una policy IAM di esempio che utilizza la chiave di condizione, consulta IAM: visualizzazione delle informazioni dell'ultimo accesso al servizio per una policy di Organizations.

Io sono: PassedToService

Lavora con operatori stringa.

Specifica il principale del servizio a cui un ruolo può essere passato. Questa chiave di condizione si applica solo all'PassRoleazione in una politica. Non può essere usata per limitare altre operazioni.

Quando si utilizza questa chiave di condizione in una policy, specificare il servizio utilizzando un principale del servizio. Il principale di un servizio è il nome di un servizio che può essere specificato nell'elemento `Principal` di una policy. Il formato tipico è `SERVICE_NAME_URL.amazonaws.com`.

Puoi utilizzare `iam:PassedToService` per limitare gli utenti in modo che possano passare ruoli solo a servizi specifici. Ad esempio, un utente potrebbe creare un ruolo di servizio che si

fida della scrittura CloudWatch di dati di log in un bucket Amazon S3 per suo conto. L'utente deve quindi collegare una policy di autorizzazione e una policy di affidabilità al nuovo ruolo di servizio. In questo caso, la policy di affidabilità deve specificare `cloudwatch.amazonaws.com` nell'elemento `Principal`. Per visualizzare una policy che consenta all'utente di trasferire il ruolo a CloudWatch, consulta. IAM: passa un ruolo IAM a un AWS servizio specifico

Utilizzando questa chiave di condizione, puoi assicurarti che gli utenti creino ruoli di servizio solo per i servizi specificati. Ad esempio, se un utente con la policy precedente tenta di creare un ruolo di servizio per Amazon EC2, l'operazione avrà esito negativo. L'errore si verifica perché l'utente non dispone dell'autorizzazione per trasferire il ruolo ad Amazon EC2.

A volte si passa un ruolo a un servizio che poi a sua volta lo passa a un servizio diverso. `iam:PassedToService` include solo il servizio finale che assume il ruolo, non il servizio intermedio che lo passa.

> **Note**
>
> Alcuni servizi non supportano questa chiave di condizione.

Sono: PermissionsBoundary

Lavora con operatori ARN.

Verifica che la policy specificata è collegata come limite delle autorizzazioni sulla risorsa del principale IAM. Per ulteriori informazioni, consulta la sezione Limiti delle autorizzazioni per le entità IAM

iam:PolicyARN

Lavora con operatori ARN.

Controlla l'Amazon Resource Name (ARN) di una policy gestita nelle richieste che implicano una policy gestita. Per ulteriori informazioni, consulta Controllo dell'accesso alle policy.

iam:ResourceTag/***nome-chiave***

Lavora con operatori stringa.

Controlla che il tag collegato alla risorsa dell'identità (utente o ruolo) corrisponda al nome e al valore della chiave specificata.

> **Note**
>
> IAM e AWS STS supportano sia la chiave di condizione `iam:ResourceTag` IAM che la chiave di condizione `aws:ResourceTag` globale.

Puoi aggiungere attributi personalizzati alle risorse IAM sotto forma di coppia chiave-valore. Per ulteriori informazioni sui tag per le risorse IAM, consulta the section called “Tagging delle risorse IAM”. Puoi utilizzare `ResourceTag` per controllare l'accesso alle risorse AWS , incluse le risorse IAM. Tuttavia, poiché IAM non supporta i tag per i gruppi, non puoi utilizzare i tag per controllare l'accesso ai gruppi.

Questo esempio mostra come creare una policy basata sull'identità che consenta di eliminare gli utenti con il tag **`status=terminated`**. Per utilizzare questa policy, sostituisci il *`testo segnaposto in corsivo`* nella policy di esempio con le tue informazioni. Quindi, segui le indicazioni fornite in Creazione di una policy o Modifica di una policy.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "iam:DeleteUser",
        "Resource": "*",
        "Condition": {"StringEquals": {"iam:ResourceTag/status": "terminated"}}
    }]
}
```

Chiavi disponibili per la federazione AWS OIDC

Puoi utilizzare la federazione OIDC per fornire credenziali di sicurezza temporanee agli utenti che sono stati autenticati tramite un provider di identità (IdP) compatibile con OpenID Connect a un provider di identità IAM OpenID Connect (OIDC) nel tuo account. AWS Esempi di tali fornitori includono GitHub Amazon Cognito, Login with Amazon e Google. È possibile utilizzare token di identità e token di accesso del proprio IdP, nonché token di account di servizio concessi ai carichi di lavoro di Amazon Elastic Kubernetes Service.

Puoi utilizzare le chiavi contestuali delle condizioni AWS OIDC per scrivere politiche che limitano l'accesso degli utenti federati alle risorse associate a un provider, un'app o un utente specifico.

Queste chiavi vengono in genere utilizzate nelle policy di trust di un ruolo. Definisci le chiavi di condizione utilizzando il nome del provider OIDC (`token.actions.githubusercontent.com`) seguito da un claim (): `:aud` **`token.actions.githubusercontent.com:aud`**

Alcune chiavi delle condizioni di federazione OIDC possono essere utilizzate nella sessione di ruolo per autorizzare l'accesso alle risorse. Se il valore è Sì nella colonna Disponibile nella sessione, è possibile utilizzare queste chiavi di condizione nelle politiche per definire a quali utenti è consentito accedere in altri servizi. AWS Quando un claim non è disponibile nella sessione, la chiave di contesto della condizione OIDC può essere utilizzata solo in una policy di fiducia dei ruoli per l'autenticazione iniziale AssumeRoleWithWebIdentity.

Seleziona il tuo IdP per vedere in che modo i claim del tuo IdP vengono mappati al contesto delle condizioni IAM. AWS

Default

L'impostazione predefinita elenca le attestazioni OIDC standard e il modo in cui vengono mappate alle chiavi di contesto AWS STS delle condizioni. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi delle AWS STS condizioni con i valori nella colonna del claim IdP JWT. Usa questa mappatura se il tuo IdP non è elencato nelle opzioni della scheda.

GitHub Actions Workflows e Google sono alcuni esempi IdPs che utilizzano l'implementazione predefinita nel token ID JWT OIDC.

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
amr	amr	Sì
aud	azp Se non è impostato alcun valore`azp`, la chiave di `aud` condizione corrisponde al `aud` reclamo.	Sì
e-mail	e-mail	No
oaud	aud	No

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
sub	sub	Sì

Per ulteriori informazioni sull'utilizzo delle chiavi contestuali delle condizioni OIDC con GitHub, vedere. Configurazione di un ruolo per il provider di GitHub identità OIDC Per ulteriori informazioni sui campi `aud` e `azp` di Google, consulta la Guida OpenID Connect di Google Identity Platform .

amr

Lavora con operatori stringa. La chiave è multivalore, il che significa che è possibile testarla in una policy con operatori di definizione di condizioni.

Esempio: `token.actions.githubusercontent.com:amr`

Il riferimento ai metodi di autenticazione include le informazioni di accesso relative all'utente. La chiave può contenere i seguenti valori:

- Se l'utente non è autenticato, la chiave contiene solo `unauthenticated`.
- Se l'utente è autenticato, la chiave contiene il valore `authenticated` e il nome del provider di accesso utilizzato nella chiamata (`accounts.google.com`).

aud

Lavora con operatori stringa.

Esempi:

- `accounts.google.com:aud`
- `token.actions.githubusercontent.com:aud`

Utilizza la chiave di `aud` condizione per verificare che il pubblico corrisponda a quello specificato nella politica. È possibile utilizzare la chiave aud con la sottochiave per lo stesso provider di identità.

Questa chiave condizionale è impostata dai seguenti campi del token:

- `aud` per gli ID client Google OAuth 2.0 dell'applicazione, quando il campo `azp` non è impostato. Quando il campo `azp` è impostato, il campo `aud` corrisponde alla chiave della condizione `accounts.google.com:oaud`.

- `azp` quando il campo `azp` è impostato. Questo può accadere per app ibride in cui un'applicazione Web e un'app Android hanno un ID client Google OAuth 2.0 diverso ma condividono lo stesso progetto delle API di Google.

Quando si scrive una policy utilizzando la chiave di condizione `accounts.google.com:aud`, occorre sapere se l'app è un'app ibrida che imposta il campo `azp`.

Campo `azp` non impostato

La policy di esempio seguente funziona per le app non ibride che non impostano il campo `azp`. In questo caso, il valore del campo `aud` del token ID di Google corrisponde a entrambi i valori della chiave di condizione `accounts.google.com:aud` e `accounts.google.com:oaud`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"Federated": "accounts.google.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "accounts.google.com:aud": "aud-value",
                    "accounts.google.com:oaud": "aud-value",
                    "accounts.google.com:sub": "sub-value"
                }
            }
        }
    ]
}
```

Campo `azp` impostato

La policy di esempio seguente funziona per app ibride che impostano il campo `azp`. In questo caso, il valore del campo `aud` del token ID di Google corrisponde solo al valore della chiave di condizione `accounts.google.com:oaud`. Il valore del campo `azp` corrisponde al valore della chiave di condizione `accounts.google.com:aud`.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Principal": {"Federated": "accounts.google.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "accounts.google.com:aud": "azp-value",
                    "accounts.google.com:oaud": "aud-value",
                    "accounts.google.com:sub": "sub-value"
                }
            }
        }
    ]
}
```

e-mail

Lavora con operatori stringa.

Esempio: `accounts.google.com:email`

Questa chiave condizionale convalida l'indirizzo e-mail dell'utente. Il valore di questa dichiarazione potrebbe non essere univoco per questo account e potrebbe cambiare nel tempo, pertanto non dovresti utilizzare questo valore come identificatore principale per verificare il tuo record utente.

oaud

Lavora con operatori stringa.

Esempio: `accounts.google.com:oaud`

Questa chiave specifica l'altro pubblico (`aud`) a cui è destinato questo token ID. Deve essere uno degli ID client OAuth 2.0 dell'applicazione.

sub

Lavora con operatori stringa.

Esempi:

- `accounts.google.com:sub`
- token.actions.githubusercontent.com: sub

Utilizza queste chiavi per verificare che l'oggetto corrisponda a quello specificato nella politica. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

Nella seguente politica sulla fiducia dei ruoli, la chiave `sub` condition limita il ruolo al GitHib ramo denominato`demo`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      "Condition": {
         "StringEquals": {
            "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
            "token.actions.githubusercontent.com:sub": "repo:octo-org/octo-
repo:ref:refs/heads/demo"
         }
      }
   ]
}
```

Amazon Cognito

Questa scheda spiega in che modo Amazon Cognito mappa le dichiarazioni OIDC per AWS STS condizionare le chiavi di contesto. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi delle AWS STS condizioni con i valori nella colonna del claim IdP JWT.

Per i ruoli utilizzati da Amazon Cognito, le chiavi vengono definite utilizzando `cognito-identity.amazonaws.com` seguita dall'attestazione.

Per ulteriori informazioni sulla mappatura dei reclami del pool di identità, consulta le mappature dei provider predefinite nella Amazon Cognito Developer Guide. Per ulteriori informazioni sulla mappatura dei reclami del pool di utenti, consulta Using the ID token nella Amazon Cognito Developer Guide.

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
amr	amr	Sì
aud	aud	Sì

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
oaud	aud	No
sub	sub	Sì

amr

Lavora con operatori stringa. La chiave è multivalore, il che significa che è possibile testarla in una policy con operatori di definizione di condizioni.

Esempio: `cognito-identity.amazonaws.com:amr`

Il riferimento ai metodi di autenticazione include le informazioni di accesso relative all'utente. La chiave può contenere i seguenti valori:

- Se l'utente non è autenticato, la chiave contiene solo `unauthenticated`.
- Se l'utente è autenticato, la chiave contiene il valore `authenticated` e il nome del provider di accesso utilizzato nella chiamata (`cognito-identity.amazonaws.com`).

Ad esempio, la seguente condizione nella politica di fiducia per un ruolo di Amazon Cognito verifica se l'utente non è autenticato.

```
"Condition": {
  "StringEquals":
    { "cognito-identity.amazonaws.com:aud": "us-east-2:identity-pool-id" },
  "ForAnyValue:StringLike":
    { "cognito-identity.amazonaws.com:amr": "unauthenticated" }
}
```

aud

Lavora con operatori stringa.

Esempio: `cognito-identity.amazonaws.com:aud`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `client_id` del token di accesso.

oaud

Lavora con operatori stringa.

Esempio — `cognito-identity.amazonaws.com:oaud`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `client_id` del token di accesso.

sub

Lavora con operatori stringa.

Esempio — `cognito-identity.amazonaws.com:sub`

L'identificatore univoco (UUID), o soggetto, dell'utente autenticato. Il nome utente potrebbe non essere univoco nel pool di utenti. La rivendicazione secondaria è il modo migliore per identificare un determinato utente. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

```
{
   "Version": "2012-10-17",
   "Statement": [
      "Condition": {
         "StringEquals": {
            "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-abcd-abcd-
abcd-123456790ab",
            "cognito-identity.amazonaws.com:sub": [
               "us-east-1:12345678-1234-1234-1234-123456790ab",
               "us-east-1:98765432-1234-1234-1243-123456790ab"
            ]
         }
      }
   ]
}
```

Login with Amazon

Questa scheda spiega in che modo Login with Amazon mappa le dichiarazioni di OIDC di AWS STS condizionare le chiavi contestuali. AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi delle AWS STS condizioni con i valori nella colonna del claim IdP JWT.

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
app_id	ID dell'applicazione	Sì
sub	ID utente	Sì
user_id	ID utente	Sì

app_id

Lavora con operatori stringa.

Esempio — `www.amazon.com:app_id`

Questa chiave specifica il contesto del pubblico che corrisponde al `aud` campo utilizzato da altri provider di identità.

sub

Lavora con operatori stringa.

Esempio: `www.amazon.com:sub`

Questa chiave verifica che l'ID utente corrisponda a quello specificato nella politica. È possibile utilizzare la chiave `sub` con la chiave `aud` per lo stesso provider di identità.

id_utente

Lavora con operatori stringa.

Esempio — `www.amazon.com:user_id`

Questa chiave specifica il contesto del pubblico che corrisponde al `aud` campo utilizzato da altri provider di identità. È possibile utilizzare la `user_id` chiave con la `id` chiave per lo stesso provider di identità.

Facebook

Questa scheda spiega in che modo Facebook mappa le chiavi contestuali dichiarate da OIDC. AWS STS AWS Puoi utilizzare queste chiavi per controllare l'accesso a un ruolo. A tale scopo, confronta le chiavi delle AWS STS condizioni con i valori nella colonna del claim IdP JWT.

AWS STS chiave di condizione	Dichiarazione IdP JWT	Disponibile in sessione
app_id	ID dell'applicazione	Sì
id	id	Sì

app_id

Lavora con operatori stringa.

Esempio — `graph.facebook.com:app_id`

Questa chiave verifica che il contesto del pubblico corrisponda al `aud` campo utilizzato da altri provider di identità.

id

Lavora con operatori stringa.

Esempio: `graph.facebook.com:id`

Questa chiave ha verificato che l'ID dell'applicazione (o del sito) corrisponda a quello specificato nella politica.

Ulteriori informazioni sulla federazione OIDC

- Guida per l'utente di Amazon Cognito
- Federazione OIDC

Chiavi disponibili per la federazione AWS STS basata su SAML

Se utilizzi una federazione basata su SAML utilizzando AWS Security Token Service (AWS STS), puoi includere chiavi di condizione aggiuntive nella politica.

Policy di affidabilità di un ruolo SAML

Nella policy di affidabilità di un ruolo è possibile includere le chiavi seguenti, che consentono di stabilire se il chiamante è autorizzato ad assumere il ruolo. Salvo per `saml:doc`, tutti i valori sono

derivati dall'asserzione SAML. Tutti gli elementi nell'elenco sono disponibili nell'editor visivo della console IAM quando crei o modifichi una policy con condizioni. Gli elementi contrassegnati con `[]` possono avere un valore che è un elenco del tipo specificato.

saml:aud

Lavora con operatori stringa.

L'URL di un endpoint a cui vengono presentate le asserzioni SAML. Il valore di questa chiave proviene dal campo `SAML Recipient` dell'asserzione, non dal campo `Audience`.

saml:commonName[]

Lavora con operatori stringa.

Questo è un attributo `commonName`.

saml:cn[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:doc

Lavora con operatori stringa.

Rappresenta il principale utilizzato per assumere il ruolo. Il formato è *`Account-ID`*/*`provider-friendly-name`*, ad esempio. `123456789012/SAMLProviderName` Il valore ID account si riferisce all'account proprietario del provider SAML.

saml:edupersonaffiliation[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonassurance[]

Lavora con operatori stringa.

Questo è un attributo `eduPerson`.

saml:edupersonentitlement[]

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonnickname[]

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonorgdn

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonorgunitdn[]

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonprimaryaffiliation

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonprimaryorgunitdn

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonprincipalname

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersonscopedaffiliation[]

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:edupersontargetedid[]

Lavora con operatori stringa.

Questo è un attributo eduPerson.

saml:eduorghomepageuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgidentityauthnpolicyuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorglegalname[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgsuperioruri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:eduorgwhitepagesuri[]

Lavora con operatori stringa.

Questo è un attributo `eduOrg`.

saml:givenName[]

Lavora con operatori stringa.

Questo è un attributo `givenName`.

saml:iss

Lavora con operatori stringa.

L'approvatore, che è rappresentato da un URN.

saml:mail[]

Lavora con operatori stringa.

Questo è un attributo `mail`.

saml:name[]

Lavora con operatori stringa.

Questo è un attributo `name`.

saml:namequalifier

Lavora con operatori stringa.

Un valore hash basato sul nome descrittivo del provider SAML. Il valore è la concatenazione dei seguenti valori, in ordine e separati da un carattere '/':

1. Il valore di risposta `Issuer` (`saml:iss`)
2. L'ID dell'account `AWS`
3. Il nome descrittivo (l'ultima parte dell'ARN) del provider SAML in IAM

La concatenazione dell'ID account e del nome descrittivo del provider SAML è disponibile per le policy IAM sotto forma di chiave `saml:doc`. Per ulteriori informazioni, consulta Identificazione univoca degli utenti nella federazione basata su SAML.

saml:organizationStatus[]

Lavora con operatori stringa.

Questo è un attributo `organizationStatus`.

saml:primaryGroupSID[]

Lavora con operatori stringa.

Questo è un attributo `primaryGroupSID`.

saml:sub

Lavora con operatori stringa.

Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).

saml:sub_type

Lavora con operatori stringa.

Questa chiave può avere il valore `persistent` o `transient` oppure consistere dell'URI `Format` completo, tratto dagli elementi `Subject` e `NameID` utilizzati nell'asserzione SAML. Il valore `persistent` indica che il valore in `saml:sub` è lo stesso per un utente da una sessione all'altra. Se il valore è `transient`, l'utente dispone di un valore `saml:sub` diverso per ogni sessione.

Per ulteriori informazioni sull'attributo `Format` dell'elemento `NameID`, consulta Configurare le asserzioni SAML per la risposta di autenticazione.

saml:surname[]

Lavora con operatori stringa.

Questo è un attributo `surnameuid`.

saml:uid[]

Lavora con operatori stringa.

Questo è un attributo `uid`.

saml: x500 [] UniqueIdentifier

Lavora con operatori stringa.

Questo è un attributo `x500UniqueIdentifier`.

Per informazioni generali sugli attributi `eduPerson` ed `eduOrg`, consulta il sito Web REFEDS. Per un elenco di `eduPerson` attributi, consulta la specifica della classe di oggetti eduPerson (201602).

Le chiavi di condizione il cui tipo è un elenco possono includere più valori. Per creare condizioni nelle policy per valori con elenchi, è possibile utilizzare gli operatori di definizione (`ForAllValues`, `ForAnyValue`). Ad esempio, per consentire l'accesso a qualsiasi utente la cui affiliazione è "facoltà" o "staff" (ma non "studente") è possibile utilizzare una condizione come la seguente:

```
"Condition": {
   "ForAllValues:StringLike": {
     "saml:edupersonaffiliation":[ "faculty", "staff"]
   }
}
```

Chiavi contestuali di federazione basate su SAML tra servizi AWS STS

Alcune chiavi di condizione di federazione basate su SAML possono essere utilizzate nelle richieste successive per autorizzare le AWS operazioni in altri servizi e chiamate. `AssumeRole` Queste sono le seguenti chiavi di condizione che possono essere utilizzate nelle politiche di fiducia dei ruoli quando i responsabili federati assumono un altro ruolo e nelle politiche delle risorse di altri AWS servizi per autorizzare l'accesso alle risorse da parte dei responsabili federati. Per ulteriori informazioni sull'utilizzo di queste chiavi, consulta Informazioni sulla federazione basata su SAML 2.0.

Seleziona una chiave di condizione per visualizzarne la descrizione.

- saml:namequalifier
- saml:sub
- saml:sub_type

> **Note**
>
> Non sono disponibili altre chiavi di condizione di federazione basate su SAML da utilizzare dopo la risposta iniziale di autenticazione del gestore dell'identità digitale esterno.

Chiavi disponibili per AWS STS

È possibile utilizzare le seguenti chiavi di condizione nelle policy di fiducia dei ruoli IAM per i ruoli che vengono assunti utilizzando le operazioni AWS Security Token Service (AWS STS).

saml:sub

Lavora con operatori stringa.

Questo è l'oggetto della richiesta, che include un valore che identifica in modo univoco un singolo utente in un'organizzazione (ad esempio, `_cbb88bf52c2510eabe00c1642d4643f41430fe25e3`).

set: AWSServiceName

Lavora con operatori stringa.

Utilizzare questa chiave per specificare il servizio in cui è possibile utilizzare un token al portatore. Quando si utilizza questa chiave di condizione in una policy, specificare il servizio utilizzando un principale del servizio. Il principale di un servizio è il nome di un servizio che può essere specificato nell'elemento `Principal` di una policy. Ad esempio, `codeartifact.amazonaws.com` è il responsabile del AWS CodeArtifact servizio.

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un token di AWS STS service bearer prima di poter accedere alle loro risorse a livello di programmazione. Ad esempio, AWS CodeArtifact richiede che le entità utilizzino token portatori per eseguire alcune operazioni. Il comando `aws codeartifact get-authorization-token` restituisce un token di connessione. È quindi possibile utilizzare il token bearer per eseguire operazioni. AWS

CodeArtifact Per ulteriori informazioni sui token del portatore, vedere Utilizzo dei token di connessione.

Disponibilità: questa chiave è presente nelle richieste che ottengono un token di connessione. Non è possibile effettuare una chiamata diretta per ottenere un token AWS STS al portatore. Quando si eseguono alcune operazioni in altri servizi, il servizio richiede il token del portatore per conto dell'utente.

È possibile utilizzare questa chiave di condizione per consentire ai principal di ottenere un token di portatore da utilizzare con un servizio specifico.

set: DurationSeconds

Lavora con operatori numerici.

Usa questa chiave per specificare la durata (in secondi) che un principale può utilizzare per ottenere un token al AWS STS portatore.

Alcuni AWS servizi richiedono l'autorizzazione per ottenere un token AWS STS service bearer prima di poter accedere alle loro risorse a livello di codice. Ad esempio, AWS CodeArtifact richiede che le entità utilizzino token portatori per eseguire alcune operazioni. Il comando `aws codeartifact get-authorization-token` restituisce un token di connessione. È quindi possibile utilizzare il token bearer per eseguire operazioni. AWS CodeArtifact Per ulteriori informazioni sui token del portatore, vedere Utilizzo dei token di connessione.

Disponibilità: questa chiave è presente nelle richieste che ottengono un token di connessione. Non è possibile effettuare una chiamata diretta per ottenere un token AWS STS al portatore. Quando si eseguono alcune operazioni in altri servizi, il servizio richiede il token del portatore per conto dell'utente. La chiave non è presente per le operazioni AWS STS di assume-role.

set: ExternalId

Lavora con operatori stringa.

Utilizza questa chiave per richiedere che un'entità principale fornisca un identificatore specifico quando assume un ruolo IAM.

Disponibilità: questa chiave è presente nella richiesta quando il principale fornisce un ID esterno mentre assume un ruolo utilizzando l' AWS API AWS CLI or.

Un identificatore univoco che può essere richiesto quando assumi un ruolo in un altro account. Se l'amministratore dell'account a cui appartiene il ruolo ha fornito un ID esterno, specifica questo valore nel parametro `ExternalId`. Questo valore può essere qualsiasi stringa, ad esempio una

passphrase o un numero di account. La funzione principale dell'ID esterno è quella di risolvere e prevenire il problema del "confused deputy" (delegato confuso). Per ulteriori informazioni sull'ID esterno e il problema del "confused deputy", consulta Come utilizzare un ID esterno per concedere l'accesso alle proprie AWS risorse a terzi.

Il valore `ExternalId` deve avere un minimo di 2 caratteri e un massimo di 1.224 caratteri. Il valore deve essere alfanumerico senza spazi. Può anche includere i seguenti simboli: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), due punti (:), barra (/) e trattino (-).

sts:RequestContext//chiave contestuale

Lavora con operatori stringa.

Utilizza questa chiave per confrontare le coppie chiave-valore del contesto di sessione incorporate nell'asserzione di contesto firmata dall'emittente del token affidabile passata nella richiesta con i valori chiave-valore del contesto specificati nella policy di attendibilità del ruolo.

Disponibilità: questa chiave è presente nella richiesta quando viene fornita un'asserzione di contesto nel parametro di `ProvidedContexts` richiesta mentre si assume un ruolo utilizzando l'operazione API. AWS STS `AssumeRole`

Questa chiave di contesto è formattata come `"sts:RequestContext/context-key":"context-value"` dove `context-key` e `context-value` rappresentano una coppia chiave-valore di contesto. Quando più chiavi di contesto sono incorporate nell'asserzione di contesto firmata passata nella richiesta, è presente una chiave di contesto per ogni coppia chiave-valore. È necessario concedere l'autorizzazione per l'azione `sts:SetContext` nella policy di attendibilità del ruolo per consentire a un principale di impostare le chiavi di contesto all'interno del token di sessione risultante. Per ulteriori informazioni sulle chiavi di contesto IAM Identity Center supportate che possono essere utilizzate con questa chiave, consulta le chiavi di AWS STS condizione per IAM Identity Center nella Guida per l'AWS IAM Identity Center utente.

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per applicare un controllo di accesso granulare in base all'utente o ai suoi attributi quando assume un ruolo. Ad esempio, puoi configurare Amazon Redshift come applicazione IAM Identity Center per accedere alle risorse Amazon S3 per conto della tua forza lavoro o delle identità federate.

La seguente policy di attendibilità del ruolo consente al principale del servizio di Amazon Redshift di assumere un ruolo nell'account `111122223333`. Concede inoltre l'autorizzazione al principale del servizio Amazon Redshift di impostare le chiavi di contesto nella richiesta, purché il set di valori della chiave di contesto `identitystore:UserId` sia `1111-22-3333-44-5555`.

Dopo aver assunto il ruolo, l'attività viene visualizzata nei AWS CloudTrail log all'interno dell'`AdditionalEventData`elemento, contenenti le coppie chiave-valore del contesto di sessione che sono state impostate dal provider del contesto nella richiesta di assunzione del ruolo. Ciò consente agli amministratori di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi. Le coppie chiave-valore vengono impostate dal provider di contesto specificato, non da o. AWS CloudTrail AWS STS Ciò consente al provider di contesto il controllo sul contesto incluso nei CloudTrail log e nelle informazioni sulla sessione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ],
            "Condition": {
                "ForAllValues:ArnEquals": {
                    "sts:RequestContextProviders": [
                        "arn:aws:iam::aws:contextProvider/IdentityCenter"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "sts:RequestContext/identitystore:UserId":
 "1111-22-3333-44-5555"
                }
            }
        }
    ]
}
```

set: RequestContextProviders

Lavora con operatori ARN.

Utilizza questa chiave per confrontare l'ARN del provider di contesto nella richiesta con l'ARN del provider di contesto specificato nella policy di attendibilità del ruolo.

Disponibilità: questa chiave è presente nella richiesta quando viene fornita un'asserzione di contesto nel parametro di `ProvidedContexts` richiesta mentre si assume un ruolo utilizzando l'operazione AWS STS `AssumeRole` API.

La condizione di esempio seguente verifica che l'ARN del provider di contesto passato nella richiesta corrisponda all'ARN specificato nella condizione della policy di attendibilità del ruolo.

```
"Condition": {
  "ForAllValues:ArnEquals": {
    "sts:RequestContextProviders": [
      "arn:aws:iam::aws:contextProvider/IdentityCenter"
    ]
  }
}
```

set: RoleSessionName

Lavora con operatori stringa.

Utilizzare questa chiave per confrontare il nome di sessione specificato da un'entità principale quando si assume un ruolo con il valore specificato nella policy.

Disponibilità: questa chiave è presente nella richiesta quando il principale assume il ruolo utilizzando il comando CLI AWS Management Console any assume-role o qualsiasi operazione API. AWS STS `AssumeRole`

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Ad esempio, è possibile richiedere che gli utenti IAM specifichino il proprio nome utente come nome di sessione. Dopo che l'utente IAM assume il ruolo, l'attività viene visualizzata nei log AWS CloudTrail con il nome della sessione corrispondente al nome utente. Ciò consente agli amministratori di distinguere tra le sessioni di ruolo quando un ruolo viene utilizzato da principali diversi.

La seguente policy di attendibilità del ruolo richiede che gli utenti IAM nell'account `111122223333` forniscano il nome utente IAM come nome di sessione quando assumono il ruolo. Questo requisito viene applicato utilizzando la variabile di condizione `aws:username` nella chiave di condizione. Questa policy consente agli utenti IAM di assumere il ruolo a cui è collegata la policy. Questa policy non consente a chiunque utilizzi credenziali temporanee di assumere il ruolo perché la variabile `username` è presente solo per gli utenti IAM.

> **Important**
>
> È possibile utilizzare qualsiasi chiave di condizione a valore singolo come variabile. Non è possibile utilizzare una chiave della condizione multi-valore come variabile.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RoleTrustPolicyRequireUsernameForSessionName",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Condition": {
                "StringLike": {"sts:RoleSessionName": "${aws:username}"}
            }
        }
    ]
}
```

Quando un amministratore visualizza il AWS CloudTrail registro di un'azione, può confrontare il nome della sessione con i nomi utente del proprio account. Nell'esempio seguente, l'utente denominato `matjac` ha eseguito l'operazione utilizzando il ruolo denominato `MateoRole`. L'amministratore può quindi contattare Mateo Jackson, che ha il nome dell'utente `matjac`.

```
    "assumedRoleUser": {
        "assumedRoleId": "AROACQRSTUVWRAOEXAMPLE:matjac",
        "arn": "arn:aws:sts::111122223333:assumed-role/MateoRole/matjac"
    }
```

Se si consente l'accesso tra account mediante i ruoli, gli utenti di un account possono assumere un ruolo in un altro account. L'ARN dell'utente assunto elencato in CloudTrail include l'account in cui esiste il ruolo. Non include l'account dell'utente che ha assunto il ruolo. Gli utenti sono univoci solo all'interno di un account. Pertanto, ti consigliamo di utilizzare questo metodo per controllare CloudTrail i log solo per i ruoli assunti dagli utenti negli account che amministri. Gli utenti potrebbero utilizzare lo stesso nome utente in più account.

set: SourceIdentity

Lavora con operatori stringa.

Utilizza questa chiave per confrontare l'identità di origine che un principale specifica quando si assume un ruolo con il valore specificato nella policy.

Disponibilità: questa chiave è presente nella richiesta quando il principale fornisce un'identità di origine assumendo un ruolo utilizzando qualsiasi comando CLI o operazione API di AWS STS assume-role. AWS STS `AssumeRole`

È possibile utilizzare questa chiave in una policy di attendibilità del ruolo per richiedere che gli utenti forniscano un nome di sessione specifico quando assumono un ruolo. Ad esempio, è possibile richiedere alla forza lavoro o alle identità federate di specificare un valore per l'identità di origine. Puoi configurare il provider di identità (IdP) per utilizzare uno degli attributi associati agli utenti, ad esempio un nome utente o un messaggio di posta elettronica come identità di origine. L'IdP passa quindi l'identità di origine come attributo nelle asserzioni o nelle affermazioni a cui invia. AWS Il valore dell'attributo di identità di origine identifica l'utente o l'applicazione che assume il ruolo.

Dopo che l'utente assume il ruolo, l'attività viene visualizzata in Log di AWS CloudTrail con il valore dell'identità di origine impostato. In questo modo è più facile per gli amministratori determinare chi o cosa ha eseguito le azioni con un ruolo. AWS Per consentire a un'identità di impostare un'identità di origine, è necessario concedere le autorizzazioni per l'operazione `sts:SetSourceIdentity`.

A differenza di `sts:RoleSessionName`, dopo aver impostato l'identità di origine, il valore non può essere modificato. È presente nel contesto della richiesta di tutte le operazioni intraprese con il ruolo dall'identità di origine. Il valore persiste nelle sessioni di ruolo successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

È possibile utilizzare la chiave di condizione `aws:SourceIdentity`globale per controllare ulteriormente l'accesso alle AWS risorse in base al valore dell'identità di origine nelle richieste successive.

La seguente policy di attendibilità del ruolo consente all'utente IAM `AdminUser` di assumere un ruolo nell'account `111122223333`. Inoltre, concede l'autorizzazione all'`AdminUser` per impostare un'identità di origine, purché il set di identità di origine sia `DiegoRamirez`.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "AllowAdminUserAssumeRole",
            "Effect": "Allow",
            "Principal": {"AWS": " arn:aws:iam::111122223333:user/AdminUser"},
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Condition": {
                "StringEquals": {"sts:SourceIdentity": "DiegoRamirez"}
            }
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo dell'identità di origine, consulta Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti.

set: TransitiveTagKeys

Lavora con operatori stringa.

Utilizzare questa chiave per confrontare le chiavi dei tag di sessione transitivi nella richiesta con quelle specificate nella policy.

Disponibilità: questa chiave è presente nella richiesta quando si effettua una richiesta utilizzando credenziali di sicurezza temporanee. Queste includono le credenziali create utilizzando qualsiasi operazione di assume-role o l'operazione `GetFederationToken`.

Quando si effettua una richiesta utilizzando credenziali di sicurezza temporanee, il contesto della richiesta include la chiave di contesto `aws:PrincipalTag`. Questa chiave include un elenco di tag di sessione, tag di sessione transitivi e tag di ruolo. I tag di sessione transitivi sono tag che persistono in tutte le sessioni successive quando si utilizzano le credenziali di sessione per assumere un altro ruolo. L'assunzione di un ruolo partendo da un altro si chiama concatenamento del ruolo.

È possibile utilizzare questa chiave di condizione in una policy per richiedere l'impostazione di specifici tag di sessione come transitivi quando si assume un ruolo o si federa un utente.

Azioni, risorse e chiavi di condizione per AWS i servizi

Ogni AWS servizio può definire azioni, risorse e chiavi di contesto di condizione da utilizzare nelle policy IAM. Per un elenco dei AWS servizi e delle relative azioni, risorse e chiavi di contesto delle condizioni, consulta Actions, resources and condition keys nel Service Authorization Reference.

Risorse per ulteriori informazioni su IAM

IAM è un prodotto ricco e troverai molte risorse per aiutarti a saperne di più su come IAM può aiutarti a proteggere le tue risorse Account AWS e le tue.

Argomenti

Identità

Consulta queste risorse per la creazione, la gestione e l'utilizzo di identità.

- Gestione delle identità nel Centro identità IAM: informazioni procedurali sulla creazione di utenti e gruppi nel Centro identità IAM.
- Identità IAM (utenti, gruppi di utenti e ruoli): una discussione approfondita su utenti, gruppi e ruoli.

Credenziali (password, chiavi di accesso e dispositivi MFA)

Consulta le seguenti guide per gestire password, chiavi di accesso e dispositivi MFA per i Account AWS tuoi e per gli utenti IAM.

- Gestione delle password degli utenti in AWS: descrive le opzioni per gestire le password per gli utenti IAM nel tuo account.
- Gestione delle chiavi di accesso per gli utenti IAM: descrive come funzionano le chiavi di accesso e come è possibile utilizzarle per effettuare chiamate programmatiche a AWS. Tuttavia, ci sono altre alternative più sicure delle chiavi di accesso che ti consigliamo di prendere in considerazione per prime. Per ulteriori informazioni, consulta Considerazioni e alternative per le chiavi di accesso a lungo termine nella guida Riferimenti generali di AWS .

- Utilizzo dell'autenticazione a più fattori (MFA) in AWS: illustra come configurare l'account e gli utenti IAM in modo da richiedere una password e un codice una tantum generato su un dispositivo prima che venga consentito l'accesso. (questo metodo viene talvolta denominato autenticazione a due fattori).

Per informazioni generali sui tipi di credenziali utilizzate per accedere ad Amazon Web Services, consulta Credenziali di sicurezza AWS nella guida Riferimenti generali di AWS .

Autorizzazioni e policy

Scopri i meccanismi interni delle policy IAM e trova i suggerimenti sui metodi migliori per assegnare le autorizzazioni:

- Policy e autorizzazioni in IAM: introduce il linguaggio della policy utilizzata per definire le autorizzazioni. Illustra il modo in cui collegare le autorizzazioni a utenti o gruppi oppure, per alcuni prodotti AWS , alle risorse stesse.
- Documentazione di riferimento degli elementi delle policy JSON IAM: fornisce descrizioni ed esempi per ciascun elemento del linguaggio delle policy.
- Convalida delle policy IAM: trova le risorse per la convalida delle policy JSON.
- Esempi di policy basate su identità IAM— Mostra esempi di politiche per attività comuni in vari AWS prodotti.
- Generatore di policy AWS: crea policy personalizzate tramite la scelta di prodotti e operazioni da un elenco.
- IAM Policy Simulator: verifica se una policy consentirebbe o negherebbe una richiesta specifica di. AWS

Federazione e delega

Puoi concedere l'accesso alle risorse in tuo Account AWS agli utenti che sono autenticati (registrati) altrove. Questi possono essere utenti IAM di un altro provider Account AWS (noti come delega), utenti autenticati con la procedura di accesso della tua organizzazione o utenti di un provider di identità Internet come Login with Amazon, Facebook, Google o qualsiasi altro provider di identità compatibile con OpenID Connect (OIDC). In questi casi, gli utenti ottengono credenziali di sicurezza temporanee per accedere alle risorse. AWS

- Tutorial IAM: Delega dell'accesso tra account AWS tramite i ruoli IAM: descrive in dettaglio la procedura per concedere l'accesso multi-account a un utente IAM di un altro Account AWS.
- Scenari comuni per le credenziali temporanee— Descrive i modi in cui gli utenti possono essere federati AWS dopo essere stati autenticati all'esterno di. AWS

IAM e altri prodotti AWS

La maggior parte dei AWS prodotti è integrata con IAM in modo da poter utilizzare le funzionalità IAM per proteggere l'accesso alle risorse di tali prodotti. Le seguenti risorse illustrano IAM e sicurezza per alcuni dei AWS prodotti più diffusi. Per un elenco completo dei prodotti che funzionano con IAM, inclusi i collegamenti per ulteriori informazioni su ciascuno, consulta AWS servizi che funzionano con IAM.

Uso di IAM con Amazon EC2

- Controllo dell'accesso alle risorse Amazon EC2: descrive come utilizzare le funzionalità IAM per consentire agli utenti di amministrare istanze, volumi e altri elementi di Amazon EC2.
- Utilizzo dei profili delle istanze— Descrive come utilizzare i ruoli IAM per fornire in modo sicuro le credenziali per le applicazioni eseguite su istanze Amazon EC2 e che richiedono l'accesso ad altri prodotti. AWS

Uso di IAM con Amazon S3

- Gestione delle autorizzazioni di accesso alle risorse Amazon S3: illustra il modello di sicurezza Amazon S3 per bucket e oggetti, che include le policy IAM.
- Scrittura di policy IAM: concessione dell'accesso a cartelle specifiche dell'utente in un bucket Amazon S3: descrive come permettere agli utenti di proteggere le proprie cartelle in Amazon S3. Per ulteriori post su Amazon S3 e IAM, seleziona il tag S3 sotto il titolo del post del blog.

Utilizzo di IAM con Amazon RDS

- Utilizzo di AWS Identity and Access Management (IAM) per gestire l'accesso alle risorse Amazon RDS: descrive come utilizzare IAM per controllare l'accesso alle istanze di database, agli snapshot del database e altro ancora.

- Un'introduzione alle autorizzazioni a livello di risorsa per RDS: descrive come utilizzare IAM per controllare l'accesso a specifiche istanze Amazon RDS.

Uso di IAM con Amazon DynamoDB

- Utilizzo di IAM per controllare l'accesso alle risorse DynamoDB: descrive come utilizzare IAM per consentire agli utenti di amministrare tabelle e indici DynamoDB.
- Il video seguente (8:55) spiega come fornire il controllo dell'accesso a singoli elementi o attributi (o entrambi) del database DynamoDB.

  Nozioni di base sul controllo granulare degli accessi per DynamoDB

Best practice generali relative alla sicurezza

Trova i consigli e le indicazioni degli esperti sui modi migliori per proteggere le tue risorse e le tue Account AWS risorse:

- Migliori pratiche per la sicurezza, l'identità e la conformità: trova risorse su come gestire la sicurezza di tutti Account AWS i prodotti, inclusi suggerimenti per l'architettura di sicurezza, l'uso di IAM, la crittografia e la sicurezza dei dati e altro ancora.
- Identity and Access Management: il AWS Well-Architected Framework ti aiuta a comprendere i concetti chiave, i principi di progettazione e le migliori pratiche architettoniche per la progettazione e l'esecuzione di carichi di lavoro nel cloud.
- Best practice per la sicurezza in IAM: fornisce suggerimenti sui modi di utilizzare IAM per proteggere l' Account AWS e le risorse.
- AWS CloudTrail Guida per l'utente: consente AWS CloudTrail di tenere traccia della cronologia delle chiamate API effettuate AWS e archiviare tali informazioni nei file di registro. Ciò consente di determinare quali utenti e account hanno effettuato l'accesso alle risorse nell'account, quando sono state effettuate le chiamate, quali operazioni sono state richieste e altro ancora.

Risorse generali

Esplora le seguenti risorse per saperne di più su IAM e AWS.

- Informazioni sul prodotto per IAM: informazioni generali su AWS Identity and Access Management .

- AWS re:Post per AWS Identity and Access Management: visita AWS re:Post per discutere di questioni tecniche relative a IAM con la AWS community.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo, SDK, toolkit IDE e strumenti a riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- AWS Support Center: l'hub per la creazione e la gestione dei casi. AWS Support Include anche collegamenti ad altre risorse utili, come forum, domande frequenti tecniche, stato di salute del servizio e AWS Trusted Advisor.
- AWS Support— La pagina web principale per informazioni su AWS Support one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contatti - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Chiamata all'API IAM utilizzando le richieste di query HTTP

Indice

È possibile accedere a IAM e ai AWS STS servizi in modo programmatico utilizzando l'API Query. Le richieste dell'API Query sono richieste HTTPS che devono contenere un parametro `Action` per indicare l'operazione da eseguire. IAM e AWS STS supporta le richieste GET e POST per tutte le azioni. Questo significa che l'API non richiede l'uso di GET per alcune operazioni e di POST per altre. Tuttavia, le richieste GET sono soggette ai limiti di dimensione di un URL. Anche può variare a seconda del browser, il limite tipico è di 2048 byte. Di conseguenza, per le richieste API Query che richiedono dimensioni maggiori, devi usare una richiesta POST.

La risposta è un documento XML. Per maggiori dettagli sulla risposta, consulta le pagine delle singole operazioni nella Documentazione di riferimento dell'API IAM o nella Documentazione di riferimento dell'API AWS Security Token Service.

> **Tip**
>
> Invece di effettuare chiamate dirette alle operazioni IAM o AWS STS API, puoi utilizzare uno degli AWS SDK. Gli AWS SDK sono costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Ruby, .NET, iOS, Android, ecc.). Gli SDK offrono un modo pratico per creare un accesso programmatico a IAM e. AWS Ad esempio, gli SDK si occupano di attività quali la firma crittografica delle richieste (vedi di seguito), la gestione degli errori e la ripetizione automatica delle richieste. Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta la pagina Tools for Amazon Web Services.

Per ulteriori informazioni sulle operazioni delle API e sugli errori, consulta la Documentazione di riferimento dell'API IAM o la Documentazione di riferimento dell'API AWS Security Token Service.

Endpoints

IAM e AWS STS ognuno di essi hanno un unico endpoint globale:

- (IAM) https://iam.amazonaws.com
- (AWS STS) https://sts.amazonaws.com

> **Note**
>
> AWS STS supporta anche l'invio di richieste agli endpoint regionali oltre all'endpoint globale. Prima di poterlo utilizzare AWS STS in una regione, devi prima attivare STS in quella regione per il tuo. Account AWS Per ulteriori informazioni sull'attivazione di regioni aggiuntive per AWS STS, consultaGestire AWS STS in un Regione AWS.

Per ulteriori informazioni sugli AWS endpoint e sulle regioni per tutti i servizi, consulta Endpoint e quote del servizio in. Riferimenti generali di AWS

HTTPS obbligatorio

L'API Query restituisce informazioni sensibili, come ad esempio le credenziali di sicurezza. Per tale ragione devi usare HTTPS per crittografare tutte le richieste API.

Firma delle richieste API IAM

Le richieste devono essere firmate usando un ID chiave di accesso e una Secret Access Key. L'utilizzo delle credenziali Utente root dell'account AWS per le attività quotidiane con IAM è fortemente sconsigliato. Puoi utilizzare le credenziali per un utente IAM oppure puoi usarle per generare credenziali di sicurezza AWS STS temporanee.

Per firmare le tue richieste API, ti consigliamo di utilizzare la versione 4 di AWS Signature. Per informazioni sull'uso di Signature Version 4, consulta Processo di firma con Signature Version 4 in Riferimento generale AWS .

In Riferimento generale AWS sono disponibili anche le informazioni sull'utilizzo di Signature Version 2.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- AWS Credenziali di sicurezza. Fornisce informazioni generali sui tipi di credenziali utilizzate per l'accesso. AWS

- Best practice per la sicurezza in IAM. Presenta un elenco di suggerimenti per l'utilizzo del servizio IAM per proteggere AWS le risorse.
- Credenziali di sicurezza temporanee in IAM. Descrive come creare e usare credenziali di sicurezza temporanee.

Cronologia dei documenti per IAM

La tabella seguente descrive i principali aggiornamenti della documentazione IAM.

Modifica	Descrizione	Data
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per l'autorizzazione a recuperare informazioni sulle politiche di utenti e ruoli IAM alle autorizzazioni a livello di servizio di Policy. AccessAnalyzerServiceRole	30 maggio 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto il supporto per l'autorizzazione a recuperare lo stato corrente dell'accesso pubblico a blocchi per gli snapshot di Amazon EC2 alle autorizzazioni a livello di servizio di Policy. AccessAnalyzer ServiceRole	23 gennaio 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto flussi e tabelle DynamoDB alle autorizzazioni a livello di servizio di Policy. AccessAnalyzer ServiceRole	11 gennaio 2024
AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto i bucket di directory Amazon S3 alle autorizzazioni a livello di servizio di Policy. AccessAnalyzer ServiceRole	1 dicembre 2023

IAMAccessAnalyzerReadOnlyAccess : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto le autorizzazioni a IAM AccessAnalyzer ReadOnly Access per consentirti di verificare se gli aggiornamenti alle tue policy garantiscono un accesso aggiuntivo. Questa autorizzazione è richiesta da Sistema di analisi degli accessi AWS IAM per eseguire i controlli delle policy sulla policy.	26 novembre 2023
Analizzatori degli accessi inutilizzati aggiunti per Sistema di analisi degli accessi AWS IAM	Sistema di analisi degli accessi AWS IAM semplific a l'ispezione degli accessi inutilizzati per guidarti verso il privilegio minimo. Sistema di analisi degli accessi AWS IAM analizza continuamente i tuoi account per identificare gli accessi inutilizzati e crea una dashboard centralizzata con i risultati.	26 novembre 2023
Controlli delle policy personalizzati aggiunti per Sistema di analisi degli accessi AWS IAM	Sistema di analisi degli accessi AWS IAM ora fornisce controlli delle policy personalizzati per verificare che le policy IAM aderiscano agli standard di sicurezza prima dell'implementazione.	26 novembre 2023

AccessAnalyzerServiceRolePolicy : autorizzazioni aggiunte	IAM Access Analyzer ha aggiunto le azioni IAM alle autorizzazioni a livello di servizio di AccessAnalyzerServiceRolePolicy per supportare le seguenti azioni: • Elencare le entità per una policy • Generare dettagli sull'ultimo accesso al servizio • Elencare le informazioni sulla chiave di accesso	26 novembre 2023
Informazioni relative all'ultimo accesso a un'operazione e supporto alla generazione di policy per oltre 60 servizi e operazioni aggiuntivi	Ora IAM supporta le informazioni relative all'ultimo accesso a un'operazione e genera policy con informazioni a livello di operazione per oltre 60 servizi aggiuntivi, insieme a un elenco delle operazioni per cui sono disponibili le informazioni relative all'ultimo accesso.	1° novembre 2023
Supporto per le informazioni relative all'ultimo accesso a un'operazione per più di 140 servizi	Ora IAM fornisce le informazioni relative all'ultimo accesso a un'operazione per oltre 140 servizi, insieme a un elenco delle operazioni per cui sono disponibili le informazioni relative all'ultimo accesso.	14 settembre 2023

Supporto per più dispositivi con autenticazione a più fattori (MFA) per utenti root e utenti IAM	Ora puoi aggiungere fino a otto dispositivi MFA per utente, tra cui le chiavi di sicurezza FIDO, password monouso (TOTP) di software con applicazioni di autentica zione virtuale o token TOTP hardware.	16 novembre 2022
Supporto di Sistema di analisi degli accessi AWS IAM per i nuovi tipi di risorse	Sistema di analisi degli accessi AWS IAM ha aggiunto il supporto per i seguenti tipi di risorse: • Snapshot del volume Amazon EBS • Repository di Amazon ECR • File system di Amazon EFS • Snapshot del database Amazon RDS • Snapshot del cluster database Amazon RDS • Argomenti di Amazon SNS	25 ottobre 2022
Deprecazione U2F e aggiornamento /FIDO WebAuthn	Sono state rimosse le menzioni di U2F come opzione MFA e sono state aggiunte informazioni sulle WebAuthn chiavi di sicurezza FIDO2 e FIDO.	31 maggio 2022

Aggiornamenti alla resilienza in IAM	Sono state aggiunte informazioni sul mantenimento dell'accesso alle credenziali IAM quando un evento interrompe la comunicazione tra Regioni AWS.	16 maggio 2022
Nuove chiavi della condizione globale per le risorse	Ora puoi controllare l'accesso alle risorse in base all'account, all'unità organizzativa (OU) o all'organizzazione che contiene le tue risorse. AWS Organizations In una policy IAM puoi utilizzare le chiavi della condizione globale `aws:ResourceAccount` , `aws:ResourceOrgID` e `aws:ResourceOrgPaths` .	27 aprile 2022
Esempi di codice per IAM che utilizzano gli AWS SDK	Sono stati aggiunti esempi di codice che mostrano come utilizzare IAM con un kit di sviluppo AWS software (SDK). Gli esempi sono suddivisi in estratti di codice che mostrano come richiamare le singole funzioni di servizio ed esempi che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.	7 aprile 2022

Aggiornamenti al diagramma di flusso della logica della valutazione delle policy	Aggiornamenti al diagramma di flusso della logica di valutazione della policy e al testo correlato nella sezione Determinare se una richiesta è consentita o rifiutata in un account.	17 novembre 2021
Aggiornamenti alle best practice di sicurezza	Sono state aggiunte informazioni sulla creazione di utenti amministrativi anziché utilizzare le credenziali utente root, sono state rimosse le best practice di utilizzo dei gruppi di utenti per assegnare le autorizzazioni agli utenti IAM ed è stato chiarito quando utilizzare le policy gestite anziché le policy inline.	5 ottobre 2021
Aggiornamenti all'argomento della logica di valutazione delle policy per le policy basate sulle risorse	Sono state aggiunte informazioni sull'impatto delle policy basate sulle risorse e dei diversi tipi pincipali nello stesso account.	5 ottobre 2021
Aggiornamenti alle chiavi a valore singolo e alle chiavi di condizione multivalore	Ora le differenze tra le chiavi a valore singolo e di condizione multivalore sono illustrate in modo più dettagliato. Il tipo di valore è stato aggiunto a ogni chiave di contesto della condizione globale AWS.	30 settembre 2021

Sistema di analisi degli accessi AWS IAM supporta i punti di accesso multi-regione di Amazon S3	Sistema di analisi degli accessi AWS IAM identific herà i bucket Amazon S3 che consentono l'accesso pubblico e tra account, inclusi quelli che utilizzano i punti di accesso multi-regione di Amazon S3.	2 settembre 2021
AWS aggiornamenti delle politiche gestite: aggiornam ento a una politica esistente	IAM Access Analyzer ha aggiornato una policy AWS gestita esistente.	2 settembre 2021
Più servizi supportati per la generazione di policy a livello di operazione	IAM Access Analyzer può generare policy IAM con informazioni sulle attività di accesso a livello di azione per servizi aggiuntivi. AWS	24 agosto 2021
Generazione di policy IAM per percorsi tra account	Ora puoi utilizzare IAM Access Analyzer per generare policy granulari basate sulla tua attività di accesso utilizzando un AWS CloudTrail percorso in un altro account, ad esempio un percorso centralizzato. AWS Organizations	18 agosto 2021

Controlli delle policy aggiuntivi per Sistema di analisi degli accessi AWS IAM	Sistema di analisi degli accessi AWS IAM ha esteso la convalida delle policy aggiungendo nuovi controlli delle policy che convalidano le condizioni incluse nelle policy IAM. Questi controlli analizzano il blocco delle condizioni nell'istruzione della policy e riportano avvisi di sicurezza, errori e suggerimenti insieme a consigli attuabili. Sistema di analisi degli accessi AWS IAM ha aggiunto i seguenti controlli delle policy: • Errore: formato principale del servizio non valido • Errore: chiave di tag mancante nella condizione • Avviso di sicurezza: nega NotAction con tag non supportato (chiave di condizione per il servizio) • Avviso di sicurezza: Rifiuta con chiave di condizione tag non supportata per il servizio • Avviso di sicurezza: chiavi di condizione abbinate mancanti • Suggerimento: consente l'utilizzo di una chiave di NotAction condizione del	29 giugno 2021

	tag non supportata per il servizio • Suggerimento: consentire con chiave di condizione e tag non supportata per il servizio	
Supporto per l'ultima operazione eseguita per più servizi	Ora è possibile visualizzare le informazioni dell'ultimo accesso dell'operazione alla console IAM relative all'ultima volta che un principale IAM ha utilizzato un'operazione per i seguenti servizi: Amazon EC2, IAM, Lambda e operazioni di gestione di Amazon S3. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare un rapporto sui dati. È possibile utilizzare queste informazioni per identificare le autorizzazioni non necessarie, in modo da perfezionare le policy IAM e aderire meglio al principio del privilegio minimo.	19 aprile 2021

Monitoraggio e controllo delle operazioni intraprese con i ruoli assunti	Gli amministratori possono configurare i ruoli IAM per richiedere che le identità passino un'identità di origine che viene registrata in AWS CloudTrail. La revisione delle informazioni sull'iden tità di origine consente agli amministratori di determina re chi o cosa ha eseguito le operazioni con le sessioni del ruolo assunto.	13 aprile 2021
Generazione di policy IAM basate sull'attività di accesso	È ora possibile utilizzar e Sistema di analisi degli accessi AWS IAM per generare policy granulari in base all'attività di accesso rilevata in AWS CloudTrail.	7 Aprile 2021
Controlli delle policy per Sistema di analisi degli accessi AWS IAM	Sistema di analisi degli accessi AWS IAM fornisce ora oltre 100 controlli delle policy con suggerimenti utili durante la creazione delle policy.	16 marzo 2021
Opzioni di convalida delle policy estese	Validazione estesa delle policy disponibile nella console IAM, nell' AWS API e AWS CLI utili zzando i controlli delle policy in IAM Access Analyzer per aiutarti a creare policy JSON sicure e funzionali.	15 marzo 2021
Tagging delle risorse IAM	È ora possibile taggare altre risorse IAM utilizzando una coppia di tag chiave-valore.	11 febbraio 2021

Policy delle password di default per gli utenti IAM	Se non imposti una politica di password personalizzata per le tue Account AWS, le password utente IAM devono ora soddisfare la politica di password predefinita. AWS	18 novembre 2020
Le pagine relative alle azioni, alle risorse e alle chiavi di condizione relative AWS ai servizi sono state spostate	Ogni AWS servizio può definire azioni, risorse e chiavi contestuali di condizione da utilizzare nelle policy IAM. Ora puoi trovare l'elenco dei AWS servizi e delle relative azioni, risorse e chiavi di contesto delle condizioni nel Service Authorization Reference.	16 Novembre 2020
Durata della sessione dei ruoli più lunga per gli utenti IAM	Gli utenti IAM possono ora avere una sessione di ruolo più lunga quando cambiano ruolo in AWS Management Console, riducendo le interruzioni dovute alla scadenza della sessione. Agli utenti viene concessa la durata massima della sessione impostata per il ruolo o il tempo rimanente nella sessione dell'utente IAM, a seconda di quale sia minore.	24 luglio 2020

Utilizzo di Service Quotas per richiedere aumenti rapidi per le entità IAM	Puoi richiedere aumenti di quota per quote IAM regolabili utilizzando la console Service Quotas. Ora, alcuni aumenti vengono approvati automaticamente in Service Quotas e sono disponibili nel tuo account in pochi minuti. Le richieste più grandi vengono inviate a. AWS Support	25 giugno 2020
Le ultime informazioni di accesso a IAM ora includono le operazioni di gestione di Amazon S3	Oltre alle informazioni sull'ultimo accesso al servizio, è ora possibile visualizzare nella console IAM le informazioni sull'ultima volta che un principale IAM ha utilizzato un'operazione Amazon S3. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare il rapporto sui dati. Il report include informazioni sui servizi e le azioni consentiti a cui i principali hanno tentato di accedere e quando. È possibile utilizzare queste informazioni per identificare le autorizzazioni non necessari e, in modo da perfezionare le policy IAM e aderire meglio al principio del privilegio minimo.	3 giugno 2020

Aggiunta del capitolo sulla sicurezza	Il capitolo sulla sicurezza ti aiuta a capire come configurare IAM e AWS STS raggiungere i tuoi obiettivi di sicurezza e conformità. Scoprirai anche come utilizzare altri servizi AWS per monitorare e proteggere le risorse IAM.	29 aprile 2020
sts: RoleSession Nome	È ora possibile scrivere una policy che concede le autorizzazioni in base al nome di sessione specificato da un'entità principale quando si assume un ruolo.	21 aprile 2020
AWS aggiornamento della pagina di accesso	Quando accedi alla pagina di AWS accesso principale, ora puoi scegliere di accedere come utente IAM Utente root dell'account AWS o come utente IAM. In questo caso, l'etichetta sulla pagina indica se è necessario fornire il proprio indirizzo e-mail dell'utente root o le informazioni sull'utente IAM. Questa documentazione include acquisizioni dello schermo aggiornate per comprendere meglio le pagine di accesso AWS .	4 marzo 2020

aws:via AWSService e aws: chiavi di condizione CalledVia	È ora possibile scrivere una policy per limitare se i servizi possono effettuare richieste per conto di un principale IAM (utente o ruolo). Quando un principale effettua una richiesta a un servizio AWS , tale servizio potrebbe utilizzare le credenziali del principale per effettuare richieste successiv e ad altri servizi. Utilizzar e la chiave di condizione `aws:ViaAWSService` per stabilire se un servizio effettua una richiesta utilizzando le credenziali di un principale. Utilizzare le chiavi di condizion e `aws:CalledVia` per stabilire se servizi specifici fanno una richiesta utilizzando le credenziali di un principale.	20 febbraio 2020
Policy Simulator aggiunge il supporto per i limiti delle autorizzazioni	È ora possibile verificar e l'effetto dei limiti delle autorizzazioni sulle entità IAM con il simulatore di policy IAM.	23 gennaio 2020

Valutazione della policy multiaccount	Ora puoi scoprire come AWS valuta le politiche per l'accesso tra account diversi. Ciò si verifica quando una risorsa in un account che concede fiducia include una policy basata sulle risorse che consente al principale in un altro account di accedere alla risorsa. La richiesta deve essere consentita in entrambi gli account.	2 gennaio 2020
Tag di sessione	Puoi ora includere tag quando assumi un ruolo o esegui la federazione di un utente in AWS STS. Quando esegui l'operazione `AssumeRole` o `GetFederationToken`, puoi passare i tag di sessione come attributi. Quando si eseguono le `AssumeRoleWithWebIdentity` operazioni `AssumeRoleWithSAML` OR, è possibile passare gli attributi delle identità aziendali a. AWS	22 novembre 2019

Controlla l'accesso per gruppi di dipendenti Account AWSAWS Organizations	Ora puoi fare riferimento alle unità organizzative (OU) AWS Organizations nelle politiche IAM. Se utilizzi Organizations per organizzare gli account in unità organizzative, prima di concedere l'accesso alle tue risorse puoi richiedere che i principali apparteng ano a un'unità organizza tiva specifica. I principali includono Utente root dell'acco unt AWS, utenti IAM e ruoli IAM. A tale scopo, specifica il percorso dell'unità organizza tiva nella chiave di condizion e `aws:PrincipalOrgPa ths` nelle tue policy.	20 novembre 2019
Ultimo ruolo utilizzato	Puoi ora visualizzare la data, l'ora e la regione in cui è stato utilizzato l'ultimo ruolo. Queste informazioni consentono inoltre di identificare i ruoli inutilizzati nel tuo account. Puoi utilizzare l' AWS API AWS Management Console, AWS CLI and per visualizzare informazioni sull'ultima volta che un ruolo è stato utilizzato.	19 novembre 2019

Aggiornamento della pagina Chiavi di contesto delle condizioni globali	Puoi ora scoprire quando ciascuna delle chiavi di condizione globali è inclusa nel contesto di una richiesta . Puoi inoltre spostarti tra le chiavi più facilmente utilizzando il sommario della pagina. Le informazioni contenute nella pagina consentono di scrivere policy più accurate. Ad esempio, se i tuoi dipendenti utilizzano la federazione con ruoli IAM, devi utilizzare la chiave `aws:userId` e non la chiave `aws:userName` . La chiave `aws:userName` si applica solo agli utenti IAM e non ai ruoli.	6 ottobre 2019

ABAC in AWS	Scopri come funziona il controllo degli accessi basato sugli attributi (ABAC) nell' AWS uso dei tag e come si confronta con il modello di autorizzazione tradizionale. AWS Utilizza il tutorial ABAC per informazioni su come creare e testare una policy che consenta ai ruoli IAM con tag del principale di accedere alle risorse con i tag corrispon denti. Questa strategia consente alle persone di visualizzare o modificare solo le AWS risorse necessarie per il proprio lavoro.	3 ottobre 2019
AWS STS GetAccessKeyInfo operazione	Puoi esaminare le chiavi di AWS accesso contenute nel codice per determinare se provengono da un account di tua proprietà. Puoi passare l'ID di una chiave di accesso utilizzando il `aws sts get-access-key-info` AWS CLI comando o l'operazione `GetAccessKeyInfo` AWS API.	24 luglio 2019

Visualizzazione delle informazioni sull'ultimo accesso al servizio Organizations in IAM	Ora puoi visualizzare le informazioni sull'ultimo accesso al servizio per un' AWS Organizations entità o una policy nella AWS Organizationssezione della console IAM. Puoi anche utilizzare l' AWS API AWS CLI or per recuperare il rapporto sui dati. Questi dati includono informazioni sui servizi consentiti a cui i principali in un account di Organizations hanno provato ad accedere e il momento in cui è stato effettuato questo tentativo . Puoi utilizzare queste informazioni per identificare le autorizzazioni non necessari e, in modo da perfezionare le policy di Organizations per aderire meglio al principio del privilegio minimo.	20 giugno 2019
Utilizzo di una policy gestita in una policy di sessione	Ora puoi trasferire fino a 10 ARN di policy gestite quando assumi un ruolo. Questo consente di limitare le autorizzazioni delle credenziali temporanee del ruolo.	7 maggio 2019

AWS STS Compatibilità regionale dei token di sessione per l'endpoint globale	Ora puoi scegliere se utilizzare i token degli endpoint globali versione 1 o versione 2. I token della versione 1 sono validi solo nelle AWS regioni disponibili per impostazione predefinita. Questi token non funzionano nelle regioni abilitate manualmente, ad esempio Asia Pacifico (Hong Kong). I token Versione 2 sono validi in tutte le regioni. Tuttavia, i token versione 2 sono più lunghi e potrebbe influenzare i sistemi utilizzati per archiviare temporaneamente i token.	26 aprile 2019
Consenti l'attivazione e la AWS disabilitazione delle regioni	Ora puoi creare una policy che consenta a un amministratore di abilitare e disabilitare la regione Asia Pacifico (Hong Kong) (ap-east-1).	24 aprile 2019

Pagina Le mie credenziali di sicurezza dell'utente IAM	Gli utenti IAM possono ora gestire le proprie credenziali nella pagina Le mie credenzia li di sicurezza. Questa AWS Management Console pagina mostra informazioni sull'acco unt come l'ID dell'account e l'ID utente canonico. Gli utenti possono anche visualizzare e modificare le password, le chiavi di accesso, i certifica ti X.509, le chiavi SSH e le credenziali Git.	24 gennaio 2019
API Access Advisor	Ora puoi utilizzare l' AWS API AWS CLI and per visualizz are le informazioni sull'ultimo accesso al servizio.	7 dicembre 2018
Tagging di utenti e ruoli IAM	È ora possibile utilizzare i tag IAM per aggiungere attributi personalizzati a un'identità (utente o ruolo IAM) utilizzan do una coppia chiave-valore di tag. È anche possibile usare i tag per controllare l'accesso di un'identità alle risorse o per controllare quali tag possono essere collegati a un'identità.	14 novembre 2018
Chiavi di sicurezza U2F	È ora possibile utilizzare chiavi di sicurezza U2F come opzione per l'autenticazione a più fattori (MFA) per l'accesso alla AWS Management Console.	25 settembre 2018

Supporto per gli endpoint Amazon VPC	Ora puoi stabilire una connessione privata tra il tuo VPC e AWS STS la regione Stati Uniti occidentali (Oregon).	31 luglio 2018
Limiti delle autorizzazioni	La nuova funzionalità permette di concedere più facilmente a dipendenti affidabili la possibili tà di gestire le autorizzazioni IAM senza concedere anche l'accesso amministrativo completo a IAM.	12 luglio 2018
Leggi: ID PrincipalOrg	La nuova chiave condizionale fornisce un modo più semplice per controllare l'accesso alle AWS risorse specificando l' AWS organizzazione dei principali IAM.	17 maggio 2018
Leggi: RequestedRegion	La nuova chiave di condizion e offre un modo più semplice per utilizzare le policy IAM per controllare l'accesso alle AWS regioni.	25 aprile 2018
Maggiore durata della sessione per i ruoli IAM	Ora un ruolo IAM può avere una sessione della durata di 12 ore.	28 marzo 2018
Flusso di lavoro aggiornato per la creazione di ruoli	Il nuovo flusso di lavoro migliora il processo di creazione di relazioni di trust e di collegamento delle autorizza zioni ai ruoli.	8 settembre 2017

Account AWS processo di accesso	L'esperienza di AWS accesso aggiornata consente sia all'utente root che agli utenti IAM di utilizzare il collegamento Accedi alla console nella home page AWS Management Console della console.	25 agosto 2017
Policy IAM di esempio	La documentazione aggiornata include oltre 30 esempi di policy.	2 agosto 2017
Best practice di IAM	Le informazioni aggiunte alla sezione Utenti della console IAM semplificano l'adozione delle best practice di IAM.	5 luglio 2017
Dimensionamento automatico delle risorse	Le autorizzazioni a livello di risorsa possono controllare l'accesso e le autorizzazioni per il dimensionamento automatico delle risorse.	16 maggio 2017
Database Amazon RDS for MySQL e Amazon Aurora	Gli amministratori del database possono associare gli utenti del database agli utenti e ai ruoli IAM e quindi gestire l'accesso degli utenti a tutte le AWS risorse da un'unica posizione.	24 Aprile 2017
Ruoli collegati al servizio	I ruoli collegati ai servizi forniscono un modo più semplice e sicuro per delegare le autorizzazioni ai servizi. AWS	19 aprile 2017

Riepiloghi delle policy	I nuovi riepiloghi delle policy semplificano la comprensione delle autorizzazioni nelle policy IAM.	23 marzo 2017

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.