

Tableau Server on Linux in the AWS Cloud

Administrator Guide

Version 2021.3; Last Updated 4/20/2023

Copyright © 2024 Tableau Software®. [Legal & Privacy](#)



Contents

- Install Tableau Server on Amazon Web Services 1**
 - Tableau Server on AWS deployment options 3
 - What You Need Before You Begin 6
 - Best Practices for Installing Tableau Server on AWS 7
 - Keeping Costs Down 7
 - Tableau Server on AWS Topology 8
 - Selecting an AWS Instance Type and Size 10
 - Typical instance types and sizes for development, test, and production environments 10
 - Recommended specifications for a single production instance 11
- Self-Deploy a Single Tableau Server on AWS 13
 - Step 1: Create a Virtual Private Cloud (VPC) 14
 - Step 2: Configure networking and security 14
 - Step 3: Launch an Amazon EC2 instance 15
 - Step 4: Create an elastic IP address for the VPC 17
 - Step 5: Log in to Amazon EC2 17
 - Step 6: Install Tableau Server 18
- Self-Deploy Tableau Server on AWS in a Distributed Environment 19
 - Step 1: Create a Virtual Private Cloud (VPC) 19
 - (Optional) Step 2: Create an AWS Directory Service for the VPC 20
 - Step 3: Deploy Three Amazon EC2 Instances 21

Step 4: Install and Configure Tableau Server	22
Step 5: Create a Load Balancer for the Tableau Server Cluster	23
Securing Tableau Server on AWS	26
Network	27
Amazon VPC	27
Security Groups	28
Client Access	28
AWS Directory Service	29
Data	29
Connecting to Data Stores in AWS	30
Connecting to Data Stores Outside of AWS	30
Encrypting Data at Rest	31
Optimizing Performance of Tableau Server on AWS	32
Performance Best Practices	32
Scaling Tableau Server on AWS	35
Load Balancing	35
High Availability	36
Troubleshooting Tableau Server on AWS	38

Install Tableau Server on Amazon Web Services

When you install Tableau on an Amazon Web Services (AWS) virtual machine (VM), you still have the advanced capabilities of Tableau in a local environment. Running Tableau on an AWS VM is an excellent choice if you want the flexibility of scaling up and scaling out without having to purchase and maintain an expensive fleet of servers. For instance, you can configure Tableau to be highly available, and integrate with all of the common enterprise applications (for example, Active Directory) that many organizations rely on.

Looking for Tableau on Windows? See [Install Tableau Server on Amazon Web Services](#).

Tableau can take advantage of many native AWS services:

- **Amazon CloudWatch:** Monitors the components of your AWS-based solution.
- **Amazon Route53:** DNS web service for connecting user requests to your AWS infrastructure.
- **AWS Certificate Manager:** Helps you deploy SSL and TLS certificates for use with AWS.
- **AWS CloudFormation:** Helps you to create and manage related AWS resources in a controlled manner using pre-defined templates.
- **Elastic Load Balancing:** You can use Elastic Load Balancing to spread Tableau requests across multiple Tableau gateways.

In addition, you can store the data you use with Tableau Server using any of the following AWS features. Tableau provides native data **connectors** that enable you to connect to data in the following AWS data sources:

- **Amazon Athena**: Interactive query service that you can use to analyze data in Amazon S3 using SQL.
- **Amazon Aurora for MySQL**: High-performance MySQL-compatible database.
- **Amazon EMR Hadoop Hive**: Managed Hadoop framework for processing large amounts of data.
- **Amazon Redshift**: Data warehousing solution for storing and mining large amounts of data.
- Amazon Relational Database Service (Amazon RDS): Relational database solution supports **Amazon Aurora for MySQL**, **PostgreSQL**, **MySQL**, **MariaDB**, **Oracle**, and **Microsoft SQL Server**.
- Amazon Simple Storage Service (Amazon S3): Cloud-based storage solution.

Tableau Server on AWS deployment options

You can self-deploy Tableau Server on an Amazon EC2 instance that you provision, deploy Tableau Server using the AWS CloudFormation templates in the Tableau Server on AWS Quick Start, or deploy Tableau Server on AWS using an AWS CloudFormation template in the AWS Marketplace. The following table shows the differences between each option.

Tableau Server	AWS Self-Deployment	AWS Quick Start	AWS Marketplace
Production ready	✓	✓	✓
Upgradable	✓	✓	✓
Install on Linux	✓	✓	✓
Install on Windows	✓	✓	✓
Scale-up	✓	✓	✓
Scale-out (add nodes)	✓	✓	✗
Active Directory support	✓	✗	✗
14-day trial license	✓	✓	✓
BYOL license	✓	✓	✓
Supports latest version of Tableau	✓	✓	✓

Each of the deployment options are described in more detail below:

- **Self-Deploy to an Amazon EC2 Instance** – Provides the most flexibility and options to customize Tableau Server to your environment. If you want to quickly launch a system use the Tableau Server on AWS Quick Start or the Tableau Server (BYOL) offering in AWS Marketplace instead.

Self-deployment of Tableau Server to an Amazon EC2 instance is recommended for development, test, and production environments that start simple but may need to scale-up or scale-out later. You can upgrade Tableau Server without having to replace your VM.

For more information about self-deployment, see [Self-Deploy a Single Tableau Server on AWS](#). For more information about deploying in a distributed environment, see [Self-Deploy Tableau Server on AWS in a Distributed Environment](#).

- **Tableau Server on AWS Quick Start** – Installs Tableau Server on a standalone Amazon EC2 instance or on a cluster of standalone Amazon EC2 instances running Microsoft Windows Server, CentOS, or Ubuntu Server using an AWS CloudFormation template. This deployment strategy allows you to go live quickly and limit your overall costs.

The Tableau Server on AWS Quick Start is recommended for development, test, and production environments that start simple but may need to scale-up or scale-out later. You can use your own license (BYOL) or a free 14-day trial version. You can upgrade Tableau Server without having to replace your VM. For more information about installing Tableau Server using the Quick Start, see the [Tableau Server on AWS Quick Start](#).

For other, specialized Tableau Server on AWS Quick Starts, see:

- [Cloud Analytics Modernization on AWS Quick Start: End-to-End Cloud Analytics with Tableau Server](#)

- [Tableau Server on AWS for Healthcare Quick Start: Tableau Server on AWS for HIPAA Workloads](#)
- **Tableau Server (BYOL) in AWS Marketplace** –Installs a single-node instance of Tableau Server (BYOL) on AWS in just a few clicks. This solution uses AWS CloudFormation scripts that deploy Tableau Server at launch, to better align with how the AWS Quick Starts work today and to ensure a uniform customer experience.

You can use a 14-day free trial or, after Tableau Server is up-and-running on the Amazon EC2 instance, you can purchase a Tableau license and activate the associated product key. For more information about AWS Marketplace, see the [AWS Marketplace](#). The Tableau Server AWS Marketplace offerings have the following limitations:

- Single machine, scale-up (vertical scaling) only.
- No Active Directory. Local authentication only. If you need Active Directory authentication, see [Self-Deploy a Single Tableau Server on AWS](#).
- May be deployed to a specific set of optimal Amazon EC2 instance types and sizes.
- Supports Windows and Linux.
- Supports the latest versions of Tableau.
- Upgrade Tableau Server without having to replace your VM.

For more information about licensing options for Tableau, search for "Licensing Overview" in the Tableau Help.

What You Need Before You Begin

Before deploying Tableau Server on AWS, you must have the following:

- An AWS account.
- An Amazon EC2 key pair.

For more information, see [Amazon EC2 Key Pairs](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

- A Tableau license.
 - For a standalone deployment, you can use a 14-day trial version of Tableau, which doesn't require the use of a product key. After 14 days, you'll need to purchase a Tableau license and activate the associated product key.
 - For a cluster-based deployment, you need a user-based license (which covers all authorized users of Tableau), a core-based license (with a minimum of 16 cores), or a Tableau subscription license. To obtain a product key, [contact sales](#).

For more information about Tableau licensing options, search for "Licensing Overview" in the Tableau Help.

- (Optional) A domain managed by Amazon Route 53.
- (Optional) An SSL certificate managed by AWS Certificate Manager in the region where you are deploying Tableau Server.
- Storage on Amazon EC2 instance (a general purpose SSD (gp2) is recommended).
- (Optional) An Elastic IP address if you aren't using ELB or Amazon Route53.

Best Practices for Installing Tableau Server on AWS

The following best practices make installing Tableau Server in the cloud a better experience.

- Search for "Plan Your Deployment" in the Tableau Server Help.
- Read [What You Need Before You Begin](#).
- Search for "Security Hardening Checklist" in the Tableau Server Help
- If you're new to the cloud, use the [AWS Free Tier](#) to get familiar with working in the cloud before you deploy Tableau Server on AWS.
- Read about and understand [network topology on AWS](#) and how to architect it.
- Use an [Amazon elastic network interface](#) to provide a static MAC address for your Amazon EC2 instance.

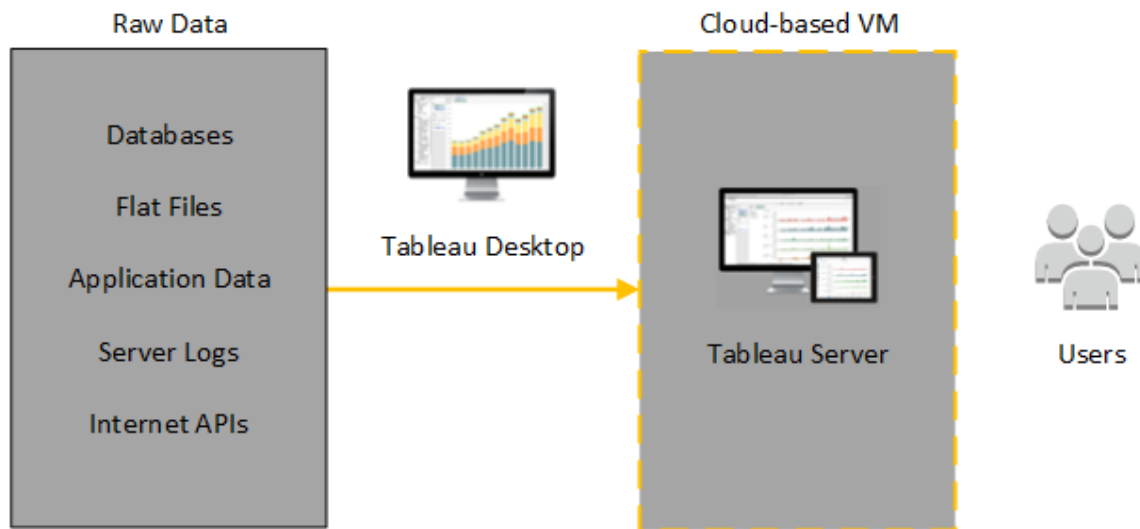
Keeping Costs Down

AWS offers cloud-based services on a pay-as-you-go basis. Costs are determined by the services you run and amount of time you use them. Different combinations of instance types and sizes have different costs. For more information about services pricing, see [AWS Pricing](#) at the AWS website. You can estimate your total monthly costs using the [AWS Simple Monthly Calculator](#) at the AWS website. You can also compare on-premises vs the cloud using the [AWS Total Cost of Ownership \(TCO\) Calculators](#) at the AWS website.

To help monitor and control usage costs on an ongoing basis, you can use Amazon CloudWatch to alert you when your monthly AWS costs reach your predefined spending threshold. For more information, see [Monitor Your Estimated Charges Using Cloudwatch](#) in the *Amazon CloudWatch User's Guide* at the AWS website.

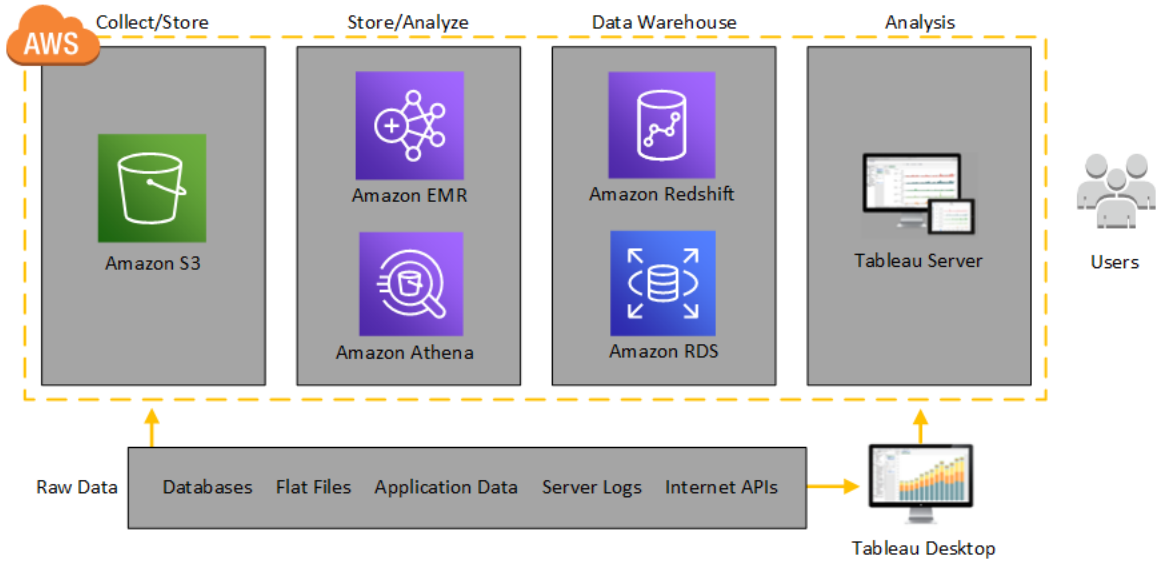
Tableau Server on AWS Topology

When deploying Tableau Server on AWS, you may choose the level and depth of integration between Tableau Desktop and Tableau Server on your Amazon EC2 instance(s). You can use Tableau Desktop and its ability to extract data from your data sources to act as a bridge between your data and Tableau Server. Based on your needs, or if you already have a lot of data in the AWS cloud, you might instead choose to leverage the full range of AWS services in conjunction with Tableau Server. The following diagram shows data source integration with Tableau Desktop and Tableau.



In the following diagram, all of your data is hosted on AWS. You can analyze both structured and unstructured data managed in a secure, scalable manner. You may leverage data that resides on AWS only, unmanaged data that lives outside of AWS, or a combination of both. This flexibility greatly increases the ability of your organization to support moves to the cloud from on-premise data, since both types of data is supported equally.

Tableau Server on Linux in the AWS Cloud Administrator Guide



Selecting an AWS Instance Type and Size

Choosing the right instance for your workload is an important factor for a successful Tableau Server deployment. You can choose from a wide range of Amazon EC2 instance types. For a complete list of all available instance types and sizes, see [Amazon EC2 Instance Types](#) at the AWS website.

At minimum, a 64-bit Tableau Server requires a 4-core CPU (the equivalent of 8 AWS vCPUs) and 16 GB RAM. However, a total of 8 CPU cores (16 AWS vCPUs) and 64GB RAM are strongly recommended for a single production Amazon EC2 instance.

An AWS vCPU is a single hyperthread of a two-thread Intel Xeon core for M5, M4, C5, C4, R4, and R4 instances. A simple way to think about this is that an AWS vCPU is equal to half a physical core. Therefore, when choosing an Amazon EC2 instance size, you should double number of cores you have purchased or wish to deploy with. Example: You have purchased an 8 core license for Tableau Server (or need to support enough active users where 8 cores are warranted). You should choose an Amazon EC2 instance type with 16 vCPUs. For more information about how many vCPUs equal a physical core, see [Physical Cores by Amazon EC2 and RDS DB Instance Type](#) at the AWS website.

Typical instance types and sizes for development, test, and production environments

- C5.4xlarge (suitable for development environments only)
- m5.4xlarge (suitable for development or testing environments only)
- r5.4xlarge (suitable for development, testing, or production environments)

Note: Installing Tableau on Amazon EC2 T2 instances is not supported.

For a performance comparison of various Amazon EC2 instance types that have been tested with Tableau Server, see [Tableau at the Speed of EC2](#).

Recommended specifications for a single production instance

Component/Resource	Amazon Web Services
CPU	16+ vCPU
Operating System	<ul style="list-style-type: none"> • Amazon Linux 2 • Red Hat Enterprise Linux (RHEL) 7.3 and higher (not 8.x), and Amazon Linux 2 • CentOS 7.3 and higher (not 8.x) • Oracle Linux 7.3 and higher (not 8.x) • The latest versions of Ubuntu 16.04 LTS and 18.04 LTS (not 17.04) <p>Note: As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.</p> <p style="text-align: center;">Earlier versions:</p>
Memory	64+ GB RAM (4GB RAM per vCPU)
Storage	<p>Two volumes:</p> <p>30-50 GiB volume for the operating system</p> <p>100 GiB or larger volume for Tableau Server</p>

Storage type	EBS recommended (SSD (gp2) or Provisioned IOPS)
Disk latency	Less than or equal to 20ms as measured by the iostat command in Linux.

Self-Deploy a Single Tableau Server on AWS

For the most flexibility in security, scaling, and capacity, you can perform a self-deployment of Tableau Server on AWS. Building a cloud-based solution has many benefits over an on-premises installation. For example, the overall total cost of ownership for building a Tableau Server solution in the cloud is normally much less than a similar on-premises solution because you don't have to buy all of the expensive hardware. In addition, the cloud can provide better uptime, reliability, and fault-tolerance, especially if you deploy your solution across different regions and Availability Zones.

When you deploy Tableau Server in a cluster in AWS, we recommend that you use Amazon Elastic Block Store (EBS) volumes with Amazon EC2. EBS provides persistent block-level storage volumes for use with Amazon EC2 instances. You should attach a separate EBS General Purpose (SSD) volume with a volume size of at least 100 GiB to allow room for installing Tableau Server. This volume size supports a baseline performance of 300 input/output operations per second (IOPS) with the ability to burst to 3,000 IOPS. For more information, see [General Purpose SSD \(gp2\) Volumes](#) in the *Amazon EC2 User Guide for Windows Instances* at the AWS website.

For heavy workloads where the storage subsystem must provide an absolute minimum number of IOPS for performance, you can either create a larger EBS General Purpose (SSD) volume to get up to 10,000 IOPS with the ratio of 3 IOPS per GiB, or use an EBS Provisioned IOPS (SSD) volume with the IOPS you need. In either case, we recommend that you use EBS-optimized instances if you plan to run heavy extracts and do a lot of data processing. For more information about Tableau Server performance, search for "Tableau Server Performance Overview" in the Tableau Server Help.

Perform the following tasks to install and configure Tableau Server in the AWS cloud.

Step 1: Create a Virtual Private Cloud (VPC)

Your first step is to create a Virtual Private Cloud (VPC) and add an [Amazon elastic network interface](#) to provide a static MAC address for your Amazon EC2 instance.

1. Sign in to AWS and then navigate to the [Amazon VPC console](https://console.aws.amazon.com/vpc/) (<https://console.aws.amazon.com/vpc/>)
2. Use the region selector to choose the location where your other AWS resources are located. Often, this is the region closest to you.
3. Under **Resources**, click **Start VPC Wizard**, on the **Step 1: Select a VPC Configuration** screen, click **VPC with a Single Public Subnet**, and then click **Select**.
4. On the **Step 2: VPC with a Single Public Subnet** screen, give your VPC a name, and then click **Create VPC**. You can leave all other settings at their default values.

Step 2: Configure networking and security

To allow inbound traffic to reach your VPC, you should limit traffic to three standard ports (HTTP, HTTPS, and RDP). For more information, see Recommended Network ACL Rules for your VPC in the Amazon Virtual Private Cloud User Guide at the AWS website.

1. Navigate to the [Amazon EC2 console](https://console.aws.amazon.com/ec2/) (<https://console.aws.amazon.com/ec2/>).
2. Use the region selector to choose the location where you created your VPC.
3. In the navigation pane, click **Security Groups**, and then click **Create Security Group**.
4. On the **Create Security Group** screen, in the **Security group name** field, enter a name for your security group.
5. In the **Description** field, enter a description for the security group.
6. In the **VPC** field, select your VPC from the list.

7. Click the **Inbound** tab, click **Add Rule**, and then in the **Type** list, select **HTTP (80)**, and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see [Adding Rules to a Security Group](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

Note: This port is required to receive incoming unencrypted web traffic.

8. Click **Add Rule**, in the **Type** list, select **HTTPS (443)**, and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see [Adding Rules to a Security Group](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

Note: This port is required to receive incoming encrypted web traffic.

9. Click **Add Rule**, in the **Type** list, select **SSH (22)**, and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see [Adding Rules to a Security Group](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

10. Click **Create**.

Step 3: Launch an Amazon EC2 instance

After you create your VPC, you can launch an Amazon EC2 instance into it.

For more information about how to launch and connect to a Linux instance, see [Getting Started with Amazon EC2 Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

1. Navigate to the [Amazon EC2 console](https://console.aws.amazon.com/ec2/) (<https://console.aws.amazon.com/ec2/>).
2. Use the region selector to choose the location where you created your VPC.
3. Under **Create Instance**, click **Launch Instance**.
4. Select an Amazon Machine Image (AMI) that meets the [system requirements](#) for Tableau Server.
5. On the **Step 2: Choose Instance Type** screen, select the instance size you want (for example, m4.2xlarge).

Note: For Tableau Server v10 on a 64-bit virtual machine, you need a minimum of 4 physical cores. On AWS, this means 8 vCPUs. For more information, see [Amazon EC2 Instance Types](#) at the AWS website.

6. Click **Next: Configure Instance Details**.
7. On the **Step 3: Configure Instance Details** screen, in the **Network** list, select your VPC.
8. Click through the steps in the wizard until you get to **Step 6: Configure Security Group**.
9. On the **Step 6: Configure Security Group** screen, click **Select an existing security group**, and then select the security group that you created earlier.
10. Click **Review and Launch**, review your configuration, and then click **Launch**.
11. When you are prompted, create a new key pair, download it as a .pem file, and keep

it in a safe place. You need the key pair in order to create a password that you can use to log into the Amazon EC2 instance.

12. When you have finished creating and downloading your key pair, click **Launch Instances**.
13. Click **View Instances** and locate your instance in the list.
14. Copy the instance ID. You'll need this in the next step.

Step 4: Create an elastic IP address for the VPC

After you launch your Amazon EC2 instance, you can create and allocate a static public IP address to your VPC.

1. Navigate to the [Amazon VPC console](https://console.aws.amazon.com/vpc/) (<https://console.aws.amazon.com/vpc/>).
2. Use the region selector to choose the location where you created your VPC.
3. In the navigation pane, click **Elastic IPs**.
4. Click **Allocate new address**, and then click **Allocate**.
5. In the **New address request succeeded** dialog box, click the Elastic IP address.
6. On the **Action** menu, click **Associate address**.
7. In **Resource type**, select **Instance**.
8. In the **Instance** drop-down list box, select your instance, and then click **Associate**.
9. Copy the new private IP address. You'll need this later.

Step 5: Log in to Amazon EC2

To run Tableau Server on AWS, you install it on the Amazon EC2 instance or instances that you just configured. To begin, you log into the Amazon EC2 instance where you want to

install Tableau Server.

Use Secure Shell (SSH) from a computer running Linux to connect to your Amazon EC2 instance running Linux. If you're connecting from a computer running Microsoft Windows, you'll need to install an SSH client, such as [PuTTY](#), to connect to your Amazon EC2 instance running Linux. For more information, see [Connecting to Your Linux Instance Using SSH](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

Step 6: Install Tableau Server

Now that you're logged in to your EC2 instance, you can install Tableau Server. The installation steps on an EC2 instance are the same as they are for any other computer. For instructions, search for "Install and Configure" in the Tableau Server Help.

After you install Tableau Server and create a Tableau Server administrator account and sign in, you can add users. For instructions, search for "Add Users to the Server" in the Tableau Server Help.

If you want to run a Tableau Server cluster, install additional Tableau Servers on additional Amazon EC2 instances. For more information, see [Self-Deploy Tableau Server on AWS in a Distributed Environment](#).

Self-Deploy Tableau Server on AWS in a Distributed Environment

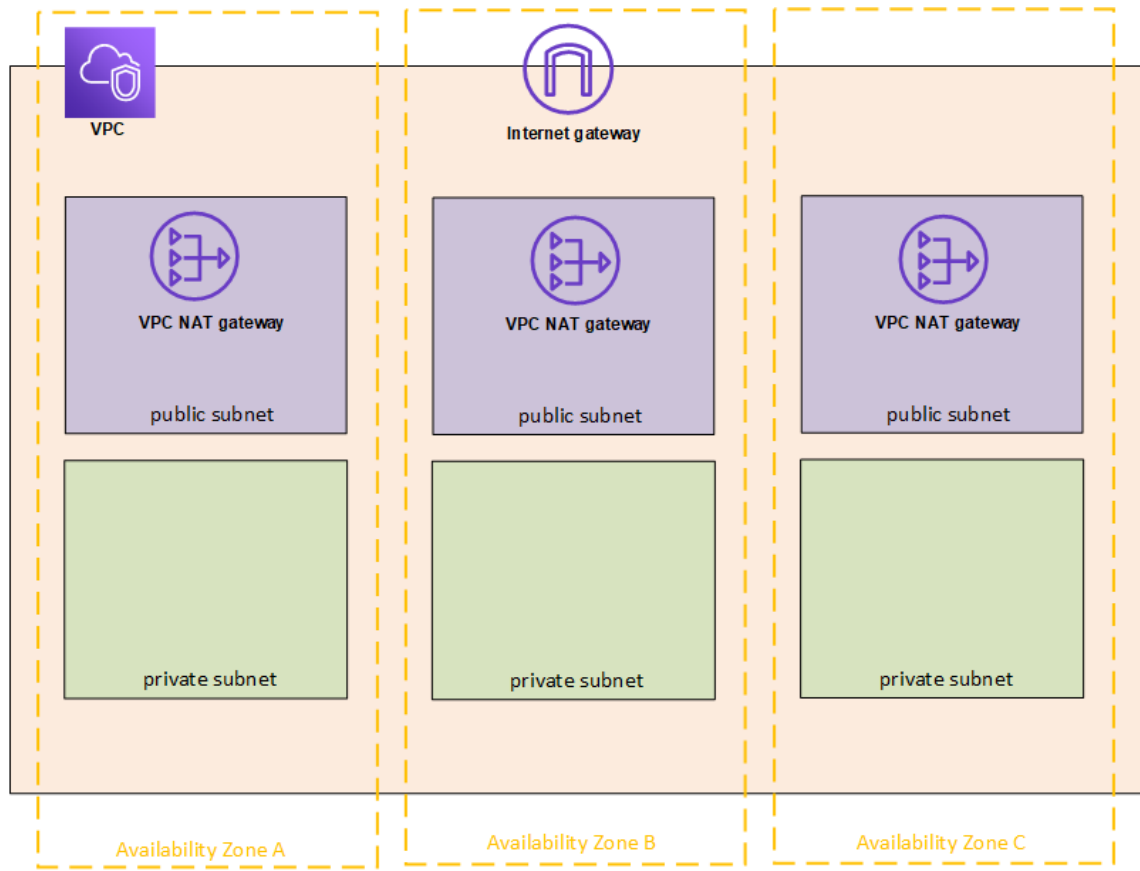
You can run Tableau Server on two virtual machines (VMs) in a distributed environment, also known as a cluster. However, if you want to run Tableau Server in a highly available (HA) distributed environment, you need to launch three or more Amazon EC2 instances of the same type and capacity to your Amazon Virtual Private Cloud (VPC) and configure them as additional nodes.

The following scenario assumes that you have three Amazon EC2 instances with Tableau Server installed on each instance. One instance is configured as the initial node, and the other two instances are configured as additional nodes.

The following steps show how to install and deploy Tableau Server on a cluster of three Amazon EC2 instances in a highly available and scalable configuration.

Step 1: Create a Virtual Private Cloud (VPC)

The following steps assume that you have an Amazon VPC with at least six subnets (three public and three private) in different Availability Zones, as shown below. Be sure to use static IP addresses when setting up your IP addresses. For information about how to create a VPC with public and private subnets, see [Scenario 2: VPC with Public and Private Subnets](#) in the *Amazon Virtual Private Cloud User Guide* at the AWS website.



Launch one EC2 instance into each of the three subnets in your VPC for use as Remote Desktop gateways.

For more information about how to launch and connect to a Linux instance, see [Getting Started with Amazon EC2 Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances* at the AWS website.

(Optional) Step 2: Create an AWS Directory Service for the VPC

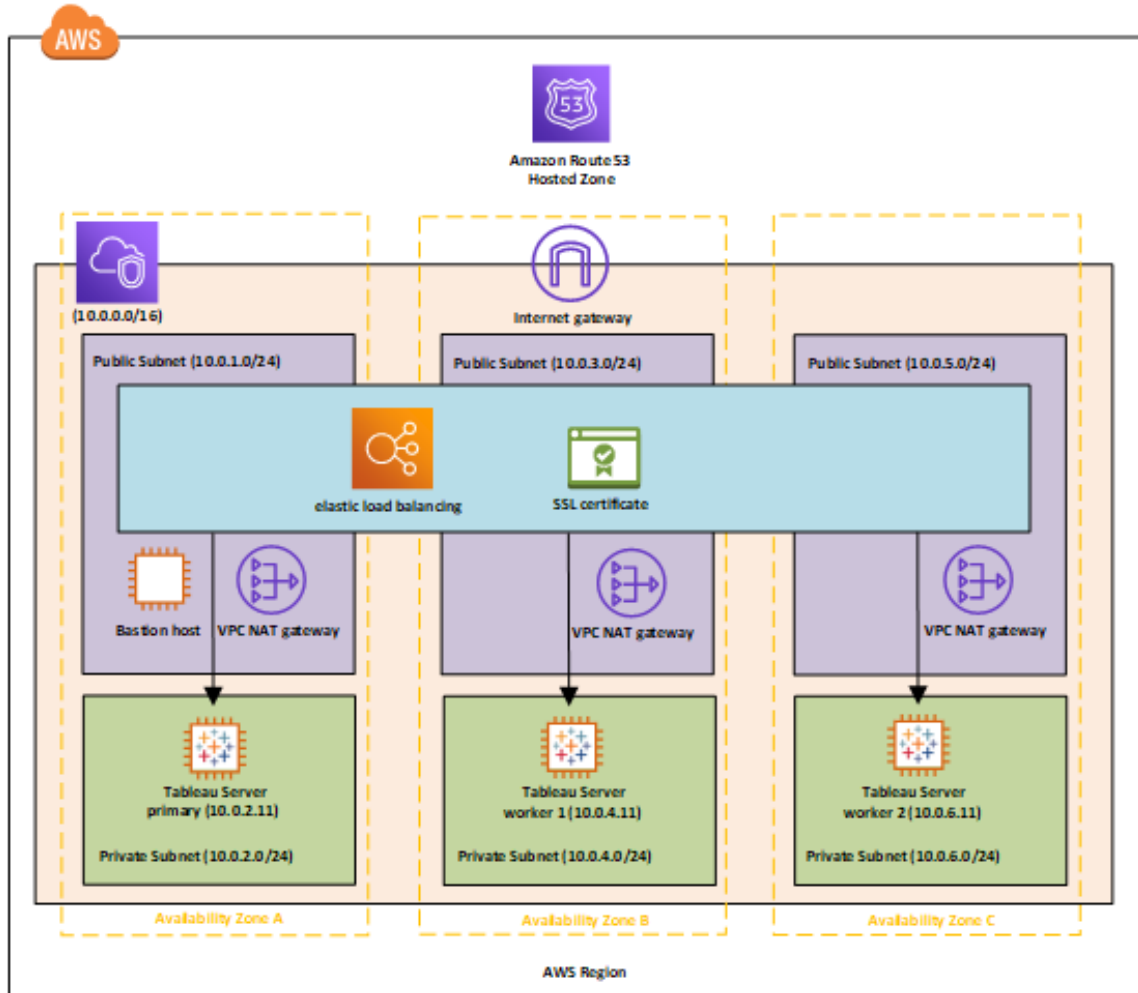
Follow the steps in the [Create a Simple AD Directory](#) section, in the *AWS Directory Service Administration Guide* at the AWS website, to create a fully-managed Samba-based directory in AWS. When you create a directory with Microsoft AD, the AWS Directory Service

creates two directory servers and DNS servers. The directory servers are created in different subnets in your Amazon VPC for redundancy, so that your directory remains accessible even if a failure occurs.

Use the two private subnets available within your Amazon VPC to create the Microsoft AD so that you can run Tableau Server across Availability Zones.

Step 3: Deploy Three Amazon EC2 Instances

Deploy three Amazon EC2 instances across three Availability Zones, as shown in the following figure. You'll install Tableau on the first node as the initial server and install Tableau on the remaining two nodes as additional servers. All of the instances should be of the same type and capacity.



After you have launched the EC2 instances, connect to them from one of the Remote Desktop Gateway (RDGW) instances by using the credentials that you decrypted for the local administrator account.

Step 4: Install and Configure Tableau Server

You'll install Tableau Server on the Amazon EC2 instances you launched in Self-Deploy Tableau Server on AWS in a Distributed Environment and configure the instances as an initial server and the rest as additional servers. For more information about installing and configuring Tableau Server on an initial server and additional servers, search for "Install

Tableau Server on Additional Nodes" in the Tableau Server Help.

Step 5: Create a Load Balancer for the Tableau Server Cluster

Follow the steps outlined in [Getting Started with Elastic Load Balancing](#) in the *Elastic Load Balancing User Guide* at the AWS website to launch a load balancer within your VPC.

1. In [Step 1: Select a Load Balancer Type](#), if you want the load balancer to be publicly accessible, select the two public subnets. Otherwise, select the **Create an internal load balancer** check box and choose the two private subnets.

If you choose to expose the load balancer with a public endpoint, make sure that you configure Elastic Load Balancing with SSL, as explained in [Create a Classic Load Balancer with an HTTPS Listener](#) in the *Elastic Load Balancing Classic Load Balancer Guide* at the AWS website.

2. In [Step 2: Configure Your Load Balancer and Listener](#) of the instructions, in the *Elastic Load Balancing User Guide* at the AWS website, ensure that your security group is configured to allow access on port 80 or 443 only, with the source limited to hosts or ranges of hosts that will access Tableau Server.
3. In [Step 4: Configure Your Target Group](#), you can specify the ping path as /.
4. In [Step 5: Register Targets with Your Target Group](#), select the Tableau Server instances and ensure that **Enable CrossZone Load Balancing** is selected so that the load balancer can load-balance the traffic across the instances in multiple Availability Zones.
5. Update Tableau Server to use the load balancer. For more information, see [Add a Load Balancer](#) in the Tableau Server Help.

Alternatively, you can configure Tableau Server to work with a load balancer by performing the following steps.

1. Create a subnet for your load balancer that has a CIDR block with a /27 bitmask, for 32 IP addresses.

Note: The IP addresses provided by Elastic Load Balancing are dynamic, and Tableau Server needs a list of static IP addresses for this configuration. To make this work, we're creating a subnet with the smallest possible CIDR range so that the IP addresses the load balancer has are limited to a finite set.

For the next steps we'll use the `tsm` command line interface, which is installed with Tableau Server by default. You can use `tsm` to perform administrative tasks from the command line on Tableau Server. For a general overview, search for "tsm Command Line Reference" in the Tableau Server Help.

2. In the Tableau Server bin directory, enter the following command, where `name` is the URL that will be used to reach Tableau Server through the load balancer:

```
tsm configuration set gateway.public.host "name"
```

3. Enter the following command, where `server1`, `server2`, and so on are the IP addresses for the given CIDR range of subnets for Elastic Load Balancing:

```
tsm configuration set gateway.trusted "server1,server2,...,server30"
```

4. Apply the changes:

```
tsm apply-pending-changes
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `-r` option, but this does not change the restart behavior. For more information, search for "`tsm pending-changes apply`" in the Tableau Server Help.

5. Start the server so the changes can take effect.

```
tsm start
```

Securing Tableau Server on AWS

Whether you deploy Tableau Server on premises or in the cloud, it's important to take steps to make your deployment secure. For information about making Tableau Server more secure, search for "Security" in the Tableau Server Help.

In addition to the security features built in to Tableau Server, AWS provides other features that you can use to help secure your Tableau Server environment, such as:

- **Amazon VPC** adds another layer of network security to your environment by creating private subnets.
- **Security Groups** determine which inbound and outbound traffic can connect to your network. Limit inbound to your IP addresses in your [Classless Inter-Domain Routing \(CIDR\)](#) block. Do not use 0000\0, which is unsecure because it allows all traffic to access your server.
- **AWS Identity and Access Management (IAM)** allows specific control over user access to features within AWS.
- **AWS Direct Connect** allows a dedicated network connection from a corporate network to AWS using industry-standard 802.1Q VLANs through an AWS Direct Connect partner. For more information, see [Requesting Cross Connects at AWS Direct Connect Locations](#) in the *AWS Direct Connect User Guide* at the AWS website.
- **Amazon EBS Encryption** offers a simple and performant way to encrypt data at rest inside your disk volumes and data-in-transit between EC2 instances and EBS storage.

You can implement enterprise application security in AWS and Tableau Server to enable a single report or dashboard to securely serve the needs of a broad and diverse user base, including both internal and external users. Enterprise application security has three main components:

Tableau Server on Linux in the AWS Cloud Administrator Guide

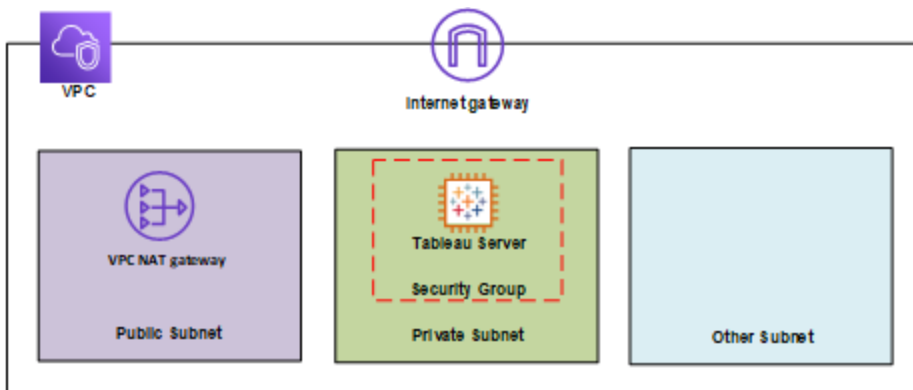
- Network
- Client Access
- Data

Network

Network security for Tableau Server in AWS relies on the use of Amazon VPC security groups with SSL for securing internal and external communications. For more information, see [Security Groups for Your VPC](#) in the Amazon Virtual Private Cloud User Guide at the AWS website.

Amazon VPC

An Amazon VPC is a distinct, isolated network within the cloud; network traffic within each Amazon VPC is isolated from all other Amazon VPCs. Using an Amazon VPC allows you to create your own network subnets and divide application layers into network subnets for a greater level of control. We recommend that you install and run Tableau Server in a separate subnet within your Amazon VPC so that you can configure the network for access to Tableau Server and other data sets. The following figure shows a typical installation of a single-node Tableau Server in an Amazon VPC.



Security Groups

Security groups enable you to define what types of network traffic can access Tableau Server. Amazon EC2 security groups act as a firewall that governs network traffic into and out of Amazon EC2 instances. You can define and assign security groups that are appropriate for your Amazon EC2 instances. By default, Amazon EC2 instances are launched with security groups that allow no inbound traffic. Before you can access your EC2 instance, you need to make changes to allow the appropriate inbound traffic.

Here are the minimum requirements for connections to Tableau Server on an EC2 instance:

- Connection via RDP (port 3389) using a Remote Desktop client to access and manage the instance and services.
- Standard web traffic via HTTP (port 80) and HTTPS (port 443), to view content hosted on, and to publish to Tableau Server.
- Communication between Tableau Server components on different instances (if any) should be allowed.

Based on these requirements, you should enable only three standard ports for inbound traffic to your EC2 instance: HTTP 80, HTTPS 443, and RDP 3389. You should also limit remote access (port 3389) from a few hosts, and also limit HTTP and HTTPS traffic to hosts within your corporate network or to a trusted set of clients.

Client Access

By default, Tableau Server uses standard HTTP requests and responses. Tableau Server can be configured for HTTPS (SSL) with customer-supplied security certificates. When Tableau Server is configured for SSL, all content and communications between clients are encrypted and use the HTTPS protocol. When you configure Tableau Server for SSL, the browser and SSL library on the server negotiate a common encryption level. Tableau

Server uses OpenSSL as the server-side SSL library, and is pre-configured to use currently accepted standards. Each web browser that accesses Tableau Server via SSL uses the standard SSL implementation provided by that browser. For more information about how Tableau Server uses SSL, search for "SSL" in the Tableau Server Help. Tableau Server will listen for SSL traffic only on port 443. You may not configure custom ports for SSL/TLS.

If you're using Elastic Load Balancing (ELB), ELB can also perform SSL termination on your behalf. Allowing ELB to handle encryption/decryption of web traffic is an easy way to secure the client's connection with Tableau Server without needing to manually configure SSL on Tableau Server itself. For more information, see [AWS Elastic Load Balancing: Support for SSL Termination](#) at the AWS website.

AWS Directory Service

Optional. The AWS Directory Service is a managed service that allows you to connect your AWS resources to an existing on-premises directory such as Microsoft Active Directory (with AD Connector), or to set up a new, stand-alone directory in the AWS cloud (with Simple AD). Connecting to an on-premises directory is easy, and after this connection is established, all users can access AWS resources and applications with their existing corporate credentials.

Using the AWS Directory Service, you can choose to use Active Directory-based authentication instead of local authentication, which creates users and assigns passwords using Tableau Server's built-in user management system. To set up Active Directory-based authentication, in the configuration step after installing Tableau Server, you must choose Active Directory. It is not possible to switch between Active Directory and local authentication later.

Data

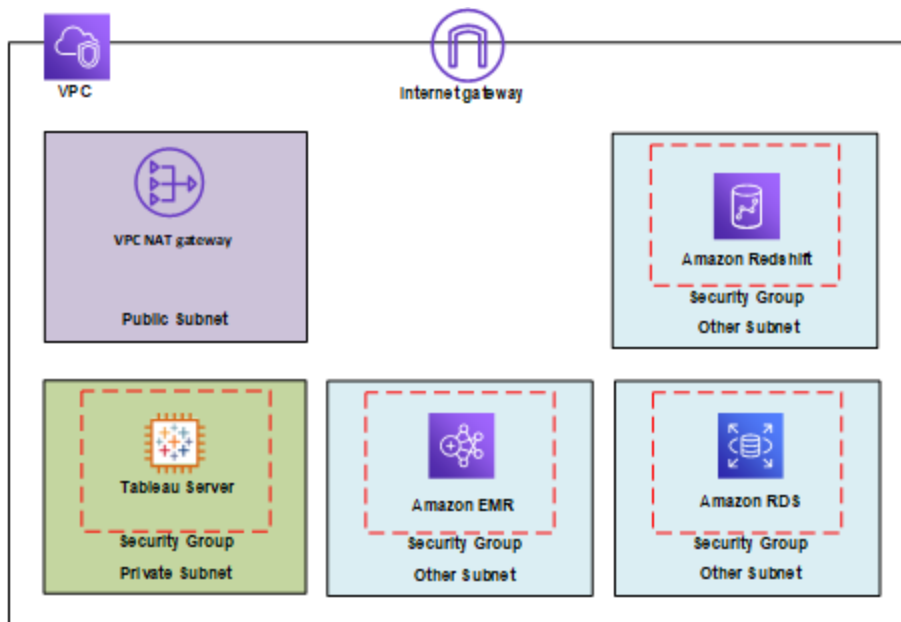
Tableau Server uses native drivers (relying on a generic ODBC adapter when native drivers are not available) to connect to databases whenever possible, for processing result sets, for refreshing extracts, and for all other communications with the database. You can configure

the driver to communicate on non-standard ports or use transport encryption, but this type of configuration is transparent to Tableau Server. However, since the Tableau Server-to-database communication is typically behind a firewall, you may choose not to encrypt this communication.

Connecting to Data Stores in AWS

You can launch AWS resources, such as Amazon Relational Database Service (Amazon RDS), Amazon Elastic MapReduce (Amazon EMR) Hadoop Hive, or Amazon Redshift, into an Amazon VPC. By placing the Tableau Server into the same Amazon VPC as your data stores, you can ensure that your traffic never leaves the Amazon VPC.

You can use subnets with security groups to launch your resources into different layers but allow them to communicate securely within an Amazon VPC, as illustrated in the following diagram.



Connecting to Data Stores Outside of AWS

You can optionally connect your Amazon VPC to your own corporate data center by using an IPsec hardware VPN connection, thus making the AWS cloud an extension of your data

center. A VPN connection consists of a virtual private gateway attached to your Amazon VPC and a customer gateway located in your data center. You might choose to use AWS Direct Connect, which is a network service that provides an alternative to using the Internet to utilize AWS cloud services. AWS Direct Connect lets you establish a dedicated network connection by using industry-standard 802.1Q VLANs through an AWS Direct Connect partner. For more information, see [Requesting Cross Connects at AWS Direct Connect Locations](#) in the *AWS Direct Connect User Guide* at the AWS website.

You can use the same connection to access public resources (such as objects stored in Amazon Simple Storage Service (Amazon S3) using public IP address space) and private resources (such as Amazon EC2 instances running within an Amazon VPC using a private IP space), while maintaining network separation between the public and private environments.

Encrypting Data at Rest

Amazon EBS encryption offers a transparent and simple way to encrypt volumes which may contain personally identifiable information (PII). EBS encryption encrypts both data at rest inside the volume and data in transit between the volume and the instance using AES-256. This feature has little-to-no impact on Tableau Server performance. Therefore, we recommend that you take advantage of this service regardless of whether your systems store PII.

Optimizing Performance of Tableau Server on AWS

Optimizing the performance of Tableau Server when it is installed on an Amazon EC2 instance in the AWS cloud adds another dimension to tuning your Tableau Server solution. This section discusses tuning Tableau Server for the cloud. For general performance tuning information, search for "Tableau Server Performance Overview" in the Tableau Server Help. For information about tools you can use to help optimize performance, search for "Performance Resources" in the Tableau Server Help.

It is important to keep in mind that every deployment of Tableau Server on AWS is different because everyone's workload is unique. Your company's employees are different, leverage different data, ask different kinds of questions, and have different business needs than other companies. For this reason, we recommend testing your Tableau Server workloads on a variety of different Amazon EC2 instance types before putting them into production. The demands of your workload will likely be influenced by the following factors:

- Heavy, light, or moderate Tableau Data Extract use
- The proportion of people who view vs. interact with visualizations and dashboards
- Tableau Data Extract refreshes which occur during or after working hours
- The number of concurrent individuals during any given period
- View and dashboard complexity
- Size of the community who leverage Tableau Web Authoring

Performance Best Practices

The following performance best practices may be useful for you to reference as you deploy Tableau on AWS:

- Always run at least 8 cores per Amazon EC2 Instance

Even with a relatively low number of users, EC2 Instances with fewer than 16 vCPUs (the equivalent of 8 cores), do not consistently perform well. For example, one r4.4xlarge instance, which has 16vCPUs, is able to handle more users, with a lower response time and error rate, than two r4.2xlarge instances, which have 8vCPUs each. This pattern is consistent as you scale up, with four 16vCPU instances and two 32vCPU instances dramatically outperforming eight 8vCPU instances.

- The workload greatly determines the results

The robustness of your workload goes a long way in determining how it performs on different EC2 instance types. Using a different set of dashboards, for example, you will see noticeable differences in performance from the same underlying virtual machine instances. Attempting to compare performance using any other workload than your own is not very useful.

- More CPU is better

CPU is often the main bottleneck for Tableau Server performance. In general, when you want to do more with Tableau, we recommend you add more and better CPU.

- Make sure your Amazon EC2 instance has enough RAM

When we ran the same workloads on instances with less CPU but more RAM, we experienced higher transactions per second (TPS), lower response time, and lower error rate. EC2 instances with too little RAM can cancel out the benefit of high-end CPU. In production, run with at least 30 GB RAM but shoot for 8 GB of RAM per core. While it is important to choose an instance with a lot of CPU, running Tableau Server on instances starved for RAM is going to lead to low performance—no matter how much CPU you have.

- You want to use SSD-based volumes, but may not need Provisioned IOPS

Tableau Server encompasses a number of processes and components, including an industrial strength database (PostgreSQL) that stores the system's metadata. Tableau Server needs a reasonable level of disk throughput in order to perform well, and we advise using Amazon Elastic Block Store (EBS) SSD- based volumes only. Magnetic disks lack the throughput needed to handle the database's requests effectively. In our test, we ran both general purpose SSD (gp2) and EBS-provisioned IOPS volumes, with two EBS disks being used for most tests. Most EBS volumes had 1500 provisioned IOPS. After running the tests again with General Purpose SSDs, our results were nearly identical with moderately demanding workloads. While there are certainly cases where Provisioned IOPS will make a noticeable difference in the performance of your Tableau Server workloads on AWS, you shouldn't assume that you need Provisioned IOPS by default. The best way to find out, of course, is to test your Tableau Server workloads yourself.

Scaling Tableau Server on AWS

Tableau Server is designed to scale up with more CPU cores and memory, and scale out when you add servers. This architecture allows you to maximize the use of compute resources while giving you the ability to scale massively.

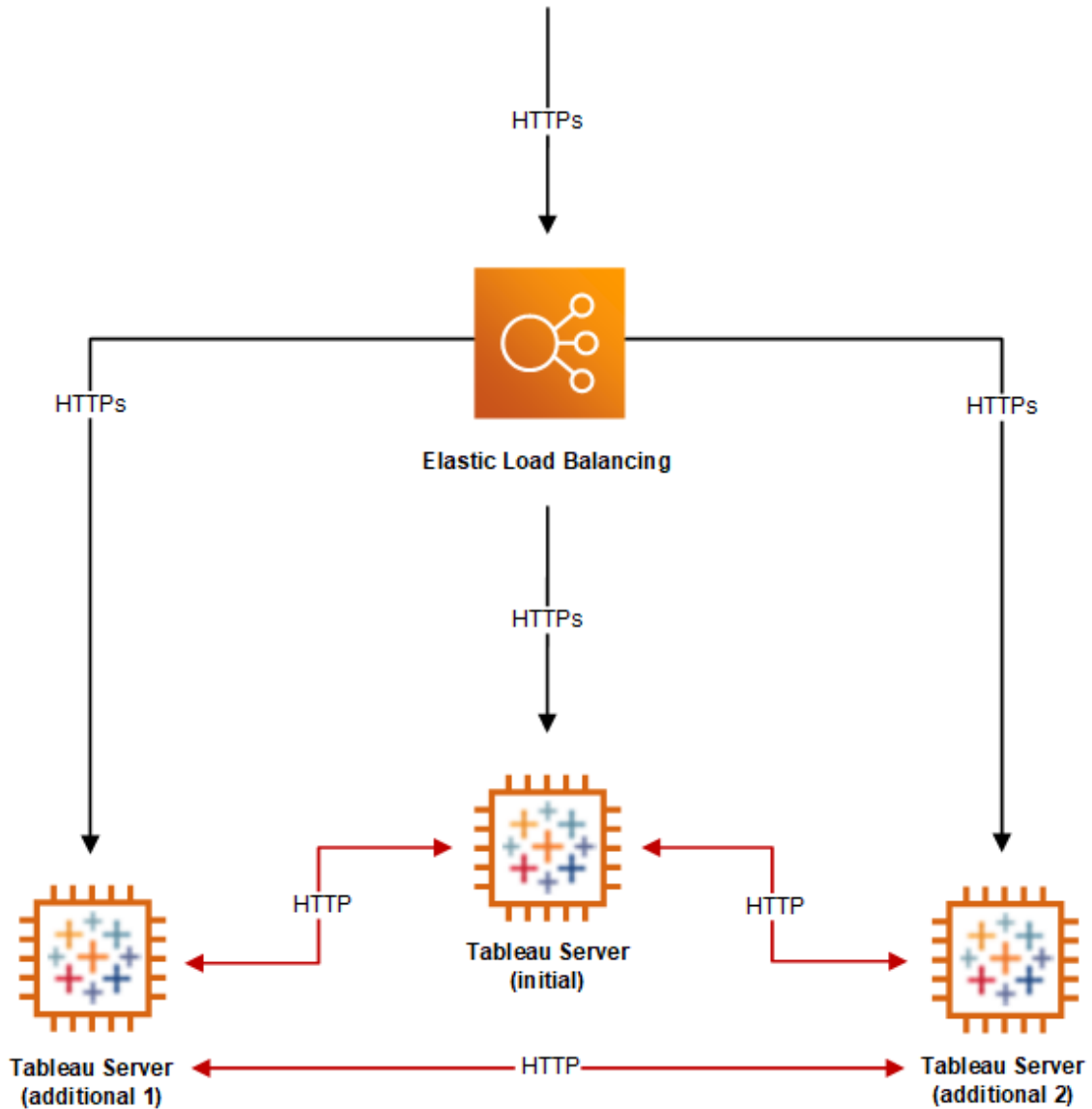
To build in redundancy, you need to add additional servers, which host copies of the repository, file store, and other processes. In this scenario, you should isolate the initial server in its own node, and it should ideally run as few of the server processes as possible.

Note: Although you can split an 8-core license onto two 4-core machines, we recommend that you scale in increments of 8-core machines only.

Load Balancing

On AWS, Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud. It enables you to achieve greater levels of fault tolerance in your applications, and seamlessly provides the required amount of load balancing capacity needed to distribute application traffic.

You can use ELB to distribute requests across multiple gateways in a Tableau Server cluster. In the figure below, all three nodes have gateways, which are used to route requests to available server processes. Unlike the repository process, there aren't any passive or standby gateway processes—all gateways are active. When you add a load balancer to a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not to the initial Tableau Server.



High Availability

Now that you have provided redundancy for the file store, repository, and gateway by adding additional nodes, you can additionally build redundancy for the initial Tableau Server. You can do this by creating a backup of the initial Tableau Server. Although the backup must be licensed during installation, it does not count as one of the three environments allowable under the Tableau end user license agreement (EULA).

Tableau Server on Linux in the AWS Cloud Administrator Guide

To configure for high availability, you need to run a failover cluster. If you configure for high availability, the initial Tableau Server may be running few or no Tableau Server processes. For more information, see [High Availability](#) in the Tableau Server Help.

Troubleshooting Tableau Server on AWS

Follow the suggestions in this topic to resolve common issues with Tableau Server when installed on an Amazon EC2 instance in the AWS cloud.

- **TSM CLI does not work from additional nodes**

The TSM cli is designed so you can run tsm commands from any node in a server cluster by including the `-s` option to specify the name or IP address of the initial node. The following symptom might occur when running TSM commands from multiple nodes in AWS:

- When running a tsm command with the `-s` option, a message can display:

```
Unable to verify the server's HTTPS certificate.
```

To run TSM at the command line from any node other than the initial node, you need to use the `-s` switch and the private IP address of the initial node.

- **Tableau Server does not use all CPU cores**

One or more of the following symptoms might occur when working with Tableau Server core-based licensing:

- When installing Tableau Server, the installer might show fewer cores than you would expect to see for the computer running Tableau Server.
- On the Licenses page on Tableau Server, the number of cores listed under Licenses in Use is less than you would expect to see.
- When you run `tsm licenses list`, Tableau Server might report fewer cores in use that you would expect to see.

For more information, see [Not All Cores Recognized by Tableau Server](#).

- **There is high I/O latency**

Changing the disk driver performance setting to **Better Performance** may give you better results.