

---

# Google Cloud Platform Customer Supplied Encryption Key: Integration Guide

---

THALES LUNA HSM AND DPOD LUNA CLOUD HSM

## Document Information

<b>Document Part Number</b>	007-013795-001
<b>Revision</b>	C
<b>Release Date</b>	4 November 2022

## Trademarks, Copyrights, and Third-Party Software

Copyright © 2022 Thales Group. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales Group and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

# CONTENTS

Overview .....	4
Certified Platforms .....	4
Prerequisites .....	5
Configure Luna HSM Device .....	5
Configure Luna Cloud HSM Service .....	6
Set up Google Cloud Platform .....	8
Before You Begin .....	9
Integrating Luna HSM with Google Cloud Platform CSEK .....	9
Generate the CSEK for Google Cloud .....	9
Create the Encrypted Contents using CSEK .....	13
Stop and Start the VM encrypted by CSEK .....	19
Appendix: Connect a VM through SSH .....	21
Contacting Customer Support .....	25
Customer Support Portal .....	25
Telephone Support .....	25

## Overview

This integration guide describes how you can use a Luna HSM partition to store the Customer Supplied Encryption Key for Google Cloud Storage. By default, Google Compute Engine uses encryption keys stored in the cloud to encrypt all data at rest and manages this encryption for you without any additional actions on your part. Keeping the encryption keys in the cloud, however, may not be in compliance with security standards. To avoid any security risks, you can control and manage these encryption keys and provide them for cloud operations whenever needed.

As an alternative to a Google-managed server-side encryption key, you can provide your own AES-256 key, encoded in standard Base64. This key is known as a customer-supplied encryption key (CSEK). If you provide your own encryption keys, Google Compute Engine uses your keys to encrypt the Google-generated keys used to encrypt and decrypt your data. Only users who provides the correct encryption key can use resources protected by a customer-supplied encryption key. If you provide a CSEK, Cloud Storage does not permanently store your key on Google's servers or otherwise manage your key. Instead, you provide your key for each Cloud Storage operation, and your key is purged from Google servers after the operation is complete. Cloud Storage stores only a cryptographic hash of the key so that future requests can be validated against the hash. Your key cannot be recovered from this hash, and the hash cannot be used to decrypt your data. If you forget or lose your encryption key, there is no way for Google to recover the key or to recover any data encrypted with the lost key.

You can apply customer-supplied encryption keys to operations on an object that reads or writes data. Operations such as deleting or listing objects can be performed without providing the encryption key. The benefits of using Thales Luna HSM with the Google Cloud Platform include:

- > Secure storage for the CSEK Keys.
- > FIPS 140-2 level 3 validated hardware to secure the keys.
- > Full life cycle management of the keys.
- > Using Cloud Services with confidence.

## Certified Platforms

This integration is certified on the following platforms.

HSM Type	Platforms Tested
Luna HSM	Windows Server 2016
Luna Cloud HSM	Windows Server 2012 R2

**Luna HSM:** Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. The Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic

**Luna Cloud HSM:** Luna Cloud HSM platform provides on-demand, cloud-based HSM and Key Management services through a simple graphical user interface. With Luna Cloud HSM, security is simple, cost effective and easy to manage because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need.

## Prerequisites

Before you proceed with the integration, complete the following tasks:

### Configure Luna HSM Device

To configure a Luna HSM device:

1. Ensure that the HSM is set up, initialized, provisioned and ready for deployment. Refer to the HSM product documentation for help.
2. Create a partition that will be later used for generating encryption keys for CSEK.
3. Create and exchange certificate between the Luna Network HSM and Client system. Register client and assign partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that the partition is successfully registered and configured. The command to see the registered partitions is:

```
C:\Program Files\SafeNet\LunaClient>lunacm.exe  
lunacm.exe (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights reserved.
```

```
Available HSMs:  
Slot Id -> 0  
Label -> TPA01  
Serial Number -> 1312109862201  
Model -> LunaSA 7.7.1  
Firmware Version -> 7.7.1  
Bootloader Version -> 1.1.2  
Configuration -> Luna User Partition With SO (PW) Key Export  
With Cloning Mode  
Slot Description -> Net Token Slot  
FM HW Status -> Non-FM  
Current Slot Id: 0
```

5. For PED-authenticated HSM, enable partition policies 22 and 23 to allow activation and auto-activation.

**NOTE:** Refer to [Luna HSM documentation](#) for detailed steps on creating NTLS connection, initializing the partitions, and assigning various user roles.

## Configure Luna HSM HA (High-Availability)

Please refer to [Luna HSM documentation](#) for HA steps and details regarding configuring and setting up two or more HSM appliances on Windows and UNIX systems. You must enable the HAOnly setting in HA for failover to work so that if primary stop functioning for some reason, all calls automatically routed to secondary till primary starts functioning again.

**NOTE:** This integration is tested in both HA and FIPS mode.

## Configure Luna Cloud HSM Service

You can configure Luna Cloud HSM Service in the following ways:

- > [Standalone Cloud HSM service using minimum client package](#)
- > [Standalone Cloud HSM service using full Luna client package](#)
- > [Luna HSM and Luna Cloud HSM service in hybrid mode](#)

**NOTE:** Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

### Standalone Cloud HSM service using minimum client package

To configure Luna Cloud HSM service using minimum client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

[Windows]

cvclient-min.zip

4. Run the **setenv** script to create a new configuration file containing information required by the Luna Cloud HSM service.

[Windows]

Right-click **setenv.cmd** and select **Run as Administrator**.

5. Run the **LunaCM** utility and verify that the Cloud HSM service is listed.

### Standalone Cloud HSM service using full Luna client package

To configure Luna Cloud HSM service using full Luna client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.

3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

```
[Windows]
cvclient-min.zip
```

4. Run the **setenv** script to create a new configuration file containing information required by the Luna Cloud HSM service.

```
[Windows]
Right-click setenv.cmd and select Run as Administrator.
```

5. Copy the server and partition certificates from the Cloud HSM service client directory to Luna client certificates directory:

Cloud HSM Certificates:

```
server-certificate.pem
partition-ca-certificate.pem
partition-certificate.pem
```

LunaClient Certificate Directory:

```
[Windows default location for Luna Client]
C:\Program Files\Safenet\Lunaclient\cert\
```

**NOTE:** Skip this step for Luna Client v10.2 or higher.

6. Open the configuration file from the Cloud HSM service client directory and copy the **XTC** and **REST** section.

```
[Windows]
crystoki.ini
```

7. Edit the Luna Client configuration file and add the **XTC** and **REST** sections copied from Cloud HSM service client configuration file.
8. Change server and partition certificates path from step 5 in **XTC** and **REST** sections. Do not change any other entries provided in these sections.

[XTC]

```
. . .
PartitionCAPath=<LunaClient_cert_directory>\partition-ca-certificate.pem
PartitionCertPath00=<LunaClient_cert_directory>\partition-certificate.pem
. . .
```

[REST]

```
. . .
SSLClientSideVerifyFile=<LunaClient_cert_directory>\server-certificate.pem
. . .
```

**NOTE:** Skip this step for Luna Client v10.2 or higher.

9. Edit the following entry from the **Misc** section and update the correct path for the **plugins** directory:

```
Misc]
PluginModuleDir=<LunaClient_plugins_directory>
```

```
[Windows Default]
```

```
C:\Program Files\Safenet\Lunaclient\plugins\
```

Save the configuration file. If you wish, you can now safely delete the extracted Cloud HSM service client directory.

10. Reset the **ChrystokiConfigurationPath** environment variable and point back to the location of the Luna Client configuration file.

```
[Windows]
```

In the Control Panel, search for "environment" and select **Edit the system environment variables**. Click **Environment Variables**. In both list boxes for the current user and system variables, edit **ChrystokiConfigurationPath** and point to the **crystoki.ini** file in the Luna client install directory.

11. Run the **LunaCM** utility and verify that the Cloud HSM service is listed. In hybrid mode, both Luna and Cloud HSM service will be listed.

**NOTE:** Follow the [Luna Cloud HSM documentation](#) for detailed steps for creating service, client, and initializing various user roles.

### Luna HSM and Luna Cloud HSM service in hybrid mode

To configure Luna HSM and Luna Cloud HSM service in hybrid mode, follow the steps mentioned under the [Standalone Cloud HSM service using full Luna client package](#) section above.

**NOTE:** Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

### Luna Cloud HSM Service in FIPS mode

The FIPS mode is enabled by default. However Cloud HSM service operates in both FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, ensure you enable the **Allow non-FIPS approved algorithms** check box when configuring your Cloud HSM service. Refer to the Mechanism List in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

## Set up Google Cloud Platform

Google Cloud Platform requires a google account. To use Google Cloud Services, you need to login to Google Cloud Console using your browser and set up your account. The URL for login to Google Cloud Services is provided below:

<https://console.cloud.google.com>

For using the Google Cloud Platform from command line, download and install the Google Cloud SDK on the working system. The Google Cloud SDK provides a set of tools for Cloud Platform. It contains Google



Cloud, gsutil, and bq, which you can use to access Google Compute Engine, Google Cloud Storage, Google BigQuery, and other products and services which can be access from the command-line.

You can run these tools interactively or in your automated scripts. The URL for downloading and setting up Cloud SDK is provided below:

<https://cloud.google.com/sdk/>

For more information, refer the Google Cloud SDK Online documentation at:

<https://cloud.google.com/sdk/docs/install-sdk>

**NOTE:** Before proceeding, ensure that CSEK feature support is available for your country. List of countries not supported for CSEK is available in [Google Cloud online documentation](#).

## Before You Begin

Make yourself familiar with disks, images, snapshots and VM instances using the documentation provided by Google Cloud Platform. To use the command-line examples provided in this guide:

1. Ensure that Luna HSM partition is accessible.
2. Ensure that OpenSSL is installed and added in to the PATH environment variables.
3. Ensure that google cloud SDK is installed and initialized with your default region and zone. You can use the following steps to install and initialize Google Cloud.

<https://cloud.google.com/sdk/docs/install-sdk>

## Integrating Luna HSM with Google Cloud Platform CSEK

---

Luna HSM provides strong physical protection of secure assets, including keys, and should be considered a best practice when using cloud. Following are the steps involved in this integration:

- > [Generate the CSEK for Google Cloud](#)
- > [Create the Encrypted Contents using CSEK](#)

## Generate the CSEK for Google Cloud

After creating the NTLS connection with HSM partition, download and import the Google Public Key on the HSM partition that will be used to wrap the Luna HSM generated AES256 key. To use the CSEK for Google Cloud with Luna HSM, follow the steps below.

1. Download the public certificate maintained by Google Compute Engine:

<https://cloud-certs.storage.googleapis.com/google-cloud-csek-ingress.pem>

Save the file in Luna Client Installation directory. This will simplify execution of other commands.

2. Open the command prompt and go to the Luna Client installation directory.

```
cd "C:\Program Files\SafeNet\LunaClient"
```

3. Extract the public key from the certificate using Open SSL:

```
C:\Program Files\SafeNet\LunaClient>openssl x509 -pubkey -noout -in google-
cloud-csek-ingress.pem > pubkey.pem
```

4. Import the extracted Public Key to HSM partition using the **cmu** utility provided with Luna Client.

```
C:\Program Files\SafeNet\LunaClient>cmu import -pubkey RSA -inputFile
pubkey.pem -label "google public key"
```

Provide the partition password when prompted.

```
C:\Program Files\SafeNet\LunaClient>cmu import -pubkey RSA -inputFile pubkey.pem -label "google public key"
Certificate Management Utility (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights reserved.

Please enter password for token in slot 0 : *****

...The RSA public key object was successfully created -> handle(62)
```

5. Run the **cmu list** command to ensure the key is imported successfully.

```
C:\Program Files\SafeNet\LunaClient>cmu list
```

Provide the partition password when prompted.

```
C:\Program Files\SafeNet\LunaClient>cmu list
Certificate Management Utility (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights reserved.

Please enter password for token in slot 0 : *****

handle=62      label=google public key
```

6. Ensure that the Public Key attributes (Encrypt, Verify, Wrap) are set to true using the **cmu** command below:

```
C:\Program Files\SafeNet\LunaClient>cmu getAttribute -handle=62
```

Here, handle refers to the key handle of the public key. Provide the partition password when prompted.

```
C:\Program Files\SafeNet\LunaClient>cmu getAttribute -handle=62
Certificate Management Utility (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights reserved.

Please enter password for token in slot 0 : *****

class=publicKey
token=true
private=true
label=google public key
keytype=RSA
subject=
id=
encrypt=false
wrap=false
verify=true
derive=false
startdate=
enddate=
modulus=a60e0ea3bca01019809738546459b6ef92bdf7d4ea363be08808bfa52cc0252e973b7b1adf8eb36588d9a63e25e0e3f94f6c6598f5e817f8
a06c23bd8c0796f98f0dd5567a2d1bcf43e9dd3f6d99c8bfe488915cd63515ac19bd22dcd31923b8e19e00efbb8381ad5e01690883ff629a9fad634a
a6966867447c28424643535734f122c0e29e8857736cb20c0a68df0ac0ce77283c70ea40e8d0835f4be62630d67ca0783c149e50dc4c51e787c3d7f5
859e03927b1a7336d1af64631aa029c848cba6128f277c436d317c672eabae06f600390110b3bbe5d044bf0c3d1d3735689d9ae8f7f73ccabd1295c5
a0f14cbb5e40f9150484e40f3ba4e6540c470315
modulusbits=2048
publicexponent=010001
local=false
modifiable=true
keystatus=
  Flags: 0x00
  Failed Key Authorization Limit: 3
```

If the attributes (Encrypt, Verify, Wrap) are not true, then set them using the command below:

```
C:\Program Files\SafeNet\LunaClient>cmu setAttribute -handle=62 -
encrypt=true -wrap=true
```

Here, handle refers to the key handle of the public key. Provide the partition password when prompted.

```
C:\Program Files\SafeNet\LunaClient>cmu setAttribute -handle=62 -encrypt=true -wrap=true
Certificate Management Utility (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights reserved.
Please enter password for token in slot 0 : *****
```

7. Create an AES256 key on Luna HSM partition that will be used to encrypt the contents on Google Cloud. To generate the key, run the **ckdemo** utility provided with Luna Client.

```
C:\Program Files\SafeNet\LunaClient>ckdemo
```

You'll be prompted to choose from the available options. Following are the choices (Numeric Values in bold) to generate an AES256 key:

```
( 1) Open Session
Enter your choice: 1

( 3) Login
Enter your choice: 3

Partition SO          [0]
Crypto Officer        [1]
Crypto User           [2]: 1
Enter PIN              : *****

(45) Simple Generate Key
Enter your choice: 45

Select type of key to generate

[ 1] DES      [ 2] DES2   [ 3] DES3           [ 5] CAST3
[ 6] Generic [ 7] RSA    [ 8] DSA     [ 9] DH     [10] CAST5
[11] RC2      [12] RC4    [13] RC5     [14] SSL3   [15] ECDSA
[16] AES      [17] SEED   [18] KCDSA-1024 [19] KCDSA-2048
[20] DSA Domain Param [21] KCDSA Domain Param
[22] RSA X9.31 [23] DH X9.42 [24] ARIA
[25] DH PKCS Domain Param [26] RSA 186-3 Aux Primes
[27] RSA 186-3 Primes [28] DH X9.42 Domain Param
[29] ECDSA with Extra Bits

> 16

Enter Key Length in bytes (16, 24, 32): 32

Enter Is Token Attribute [0-1]: 1
```

```

Enter Is Sensitive Attribute [0-1]: 1
Enter Is Private Attribute [0-1]: 1
Enter Encrypt Attribute [0-1]: 1
Enter Decrypt Attribute [0-1]: 1
Enter Sign Attribute [0-1]: 1
Enter Verify Attribute [0-1]: 1
Enter Wrap Attribute [0-1]: 1
Enter Unwrap Attribute [0-1]: 1
Enter Derive Attribute [0-1]: 1
Enter Extractable Attribute [0-1]: 1
Generated AES Key:          139 (0x0000008b)

```

Here, 139 is handle of generated AES Key.

- Wrap your key using the public key provided via the certificate that is managed by Compute Engine. Wrap your key using **OAEP** padding only. To wrap the key, use the same **CKDEMO** session and provide the choices (Numeric Values in bold).

```
(60) Wrap key
```

```
Enter your choice: 60
```

```

[1]DES-ECB      [2]DES-CBC      [3]DES3-ECB      [4]DES3-CBC
                  [7]CAST3-ECB    [8]CAST3-CBC
[9]RSA          [10]TRANSLA     [11]DES3-CBC-PAD [12]DES3-CBC-PAD-IPSEC
[13]SEED-ECB   [14]SEED-CBC    [15]SEED-CBC-PAD [16]DES-CBC-PAD
[17]CAST3-CBC-PAD [18]CAST5-CBC-PAD [19]AES-ECB      [20]AES-CBC
[21]AES-CBC-PAD [22]AES-CBC-PAD-IPSEC 23]ARIA-ECB      [24]ARIA-CBC
[25]ARIA-CBC-PAD [26]RSA_OAEP    [27]SET_OAEP     [28]AES-CTR
[29]DES3-CTR     [30]AES-KW      [31]AES-KWP      [34]AES-KEY-WRAP
[35]AES-GCM

```

```
Select mechanism for wrapping: 26
```

```
Enter filename of OAEP Source Data [0 for none]: 0
```

```
Enter handle of wrapping key (0 to list available objects) : 62
```

```
Enter handle of key to wrap (0 to list available objects) : 139
```

```
Wrapped key was saved in file wrapped.key
```

Here, 62 and 139 are the handles of Google Public Key and generated AES256 key, respectively.

**NOTE:** wrapped.key is the output file that contains the RSA wrapped AES key.

- Exit from **ckdemo** session by providing the choice as 0.

Enter your choice: 0

Exiting GESC SIMULATION LAB

- Encode your RSA-wrapped key in **base64** using the following Open SSL command:

```
C:\Program Files\SafeNet\LunaClient>openssl enc -base64 -in wrapped.key >
rsawrapencodedkey.txt
```

- Open the `rsawrapencodedkey.txt` file in any editor, ensure that the complete key is present in the single line, and remove any new Line Feed/Carriage Return.

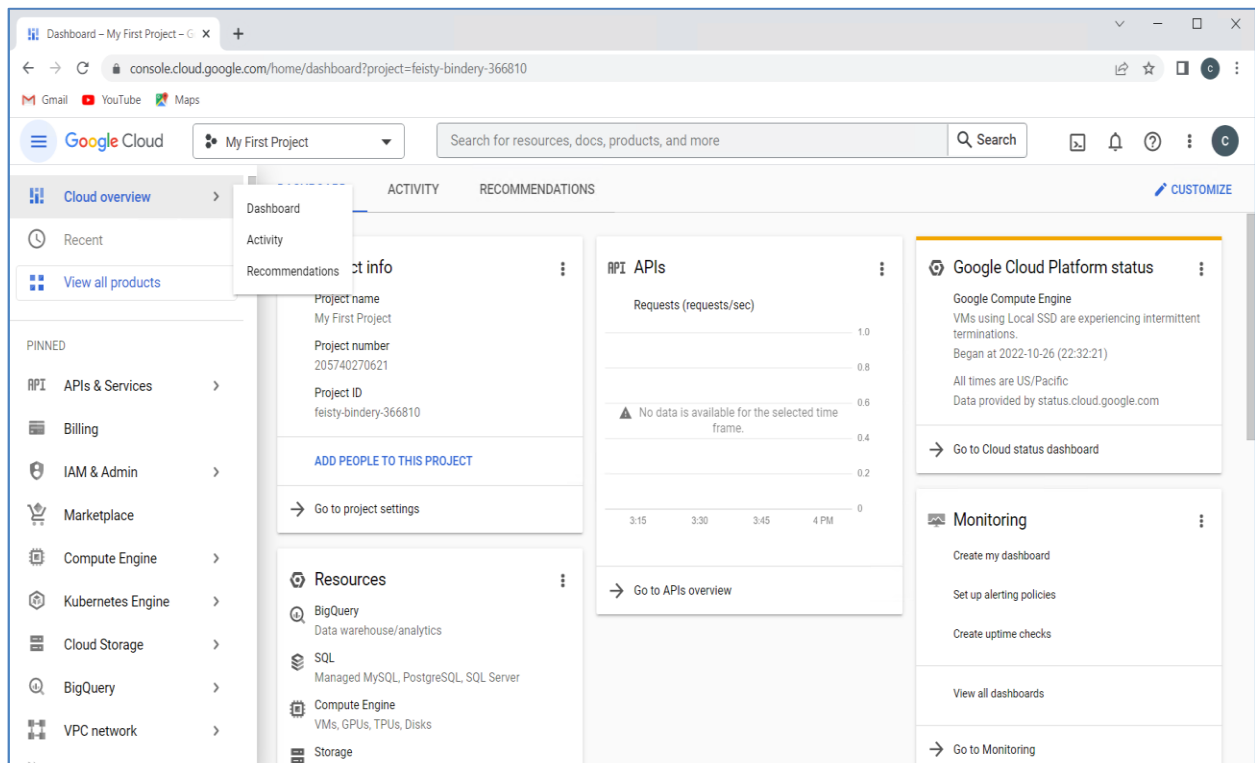
```
nfCBUW+dB7eJQb17Jb0L/EPmNoJE2aFt7+Muh7zJIrdkG/c6VJS0hiKg+T4nAroHKK10pII42aDBI0ntMMxF10zlyc4ejVmoMJ7nBgJQM0gIYwb3Hx1qBUby+PHTaOVT
5pdADXdzmMFYw6g0bf2ANz2zIMoVQZkQyUoq9RIVAMS/DNoL6AuDVjcxRE2ZTakLV/97KbYEyc8Cqt9dTMkvud0gI0mIdDwh3Cy8s2bs1j16+BxpMisIwTa/UFwW81
DGuAeOdUwLurSUIDKu0WgZrChxwDUCQxo/g1VKfrcvtFY0crXtvbuL03+26j0KVXngpGmQ+aUP27s2XYEY2zhw==
```

## Create the Encrypted Contents using CSEK

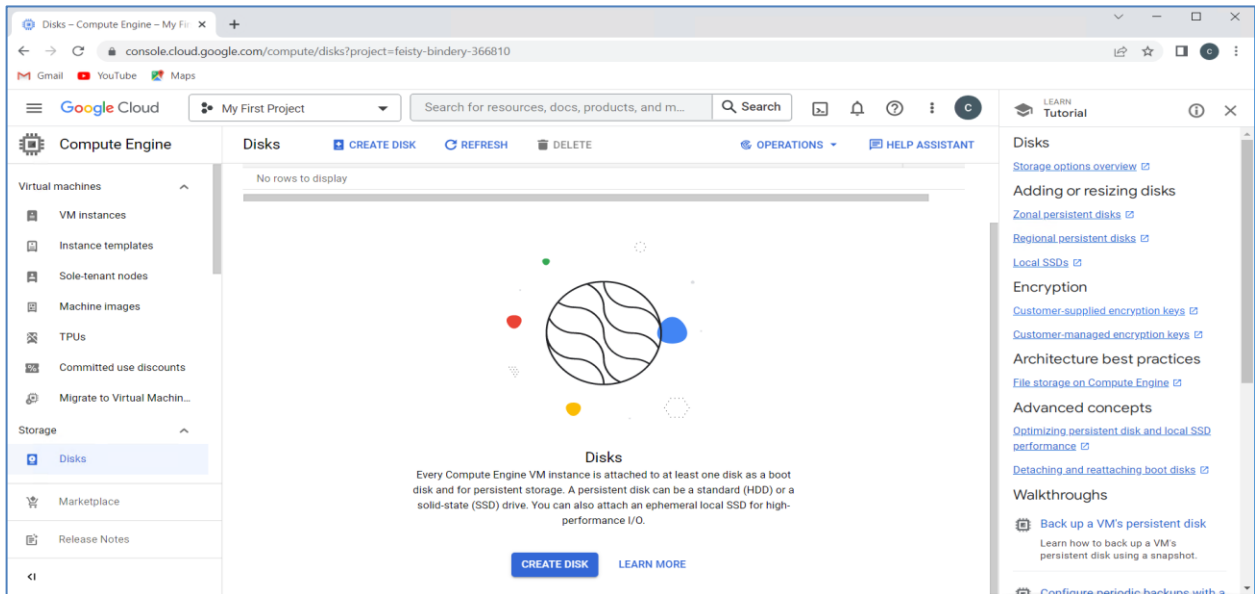
This guide demonstrates the creation of encrypted VM using console and Google Cloud tool provided by google.

- Log on to the Google Cloud Console using the below URL.

<https://console.cloud.google.com>

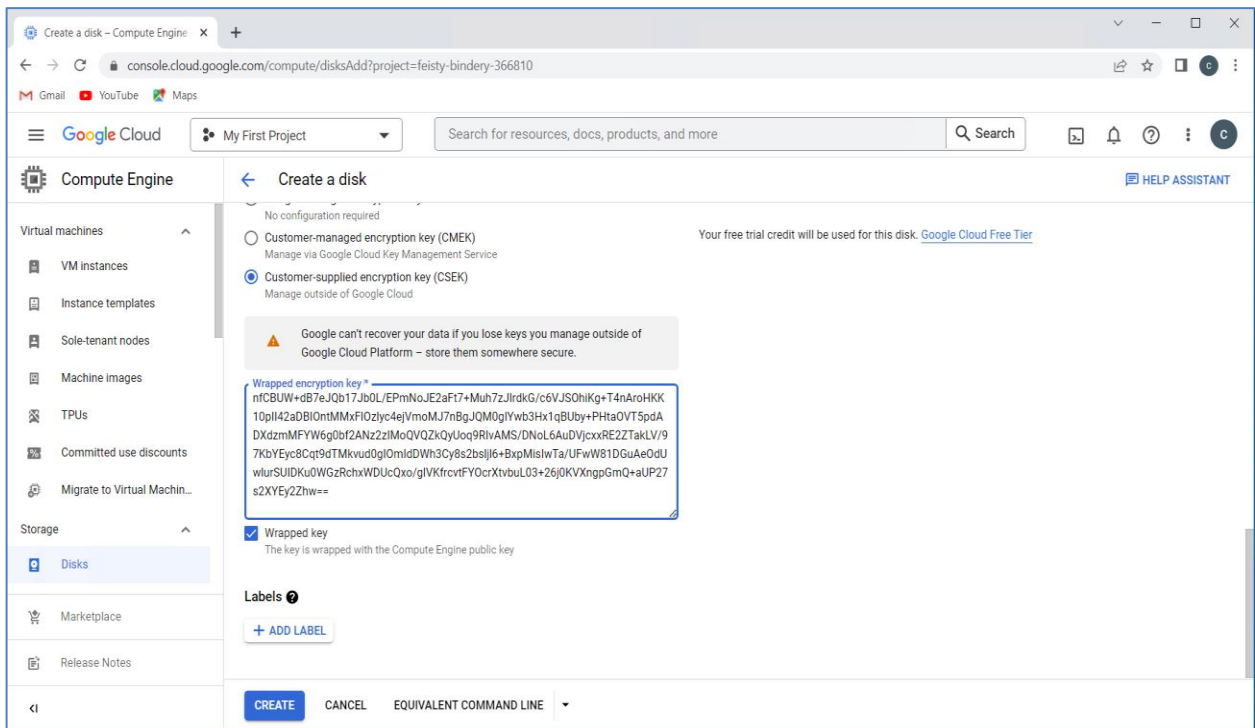


2. Click **Compute Engine -> Disks -> Create disk.**

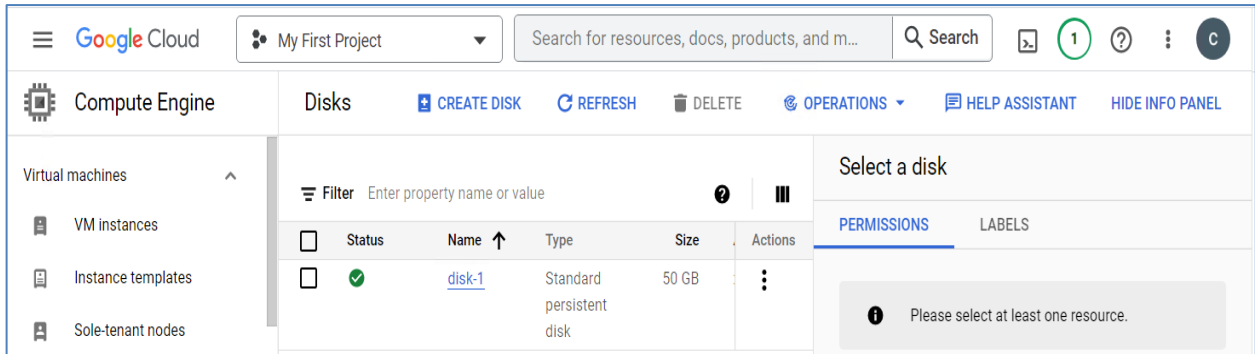


3. Enter the **Name**, **Description**, select **Zone** and **Disk Type** as **Standard persistent disk**. Select **Source type**, **Source Image** (OS that need to be installed), and **Size (GB)**.

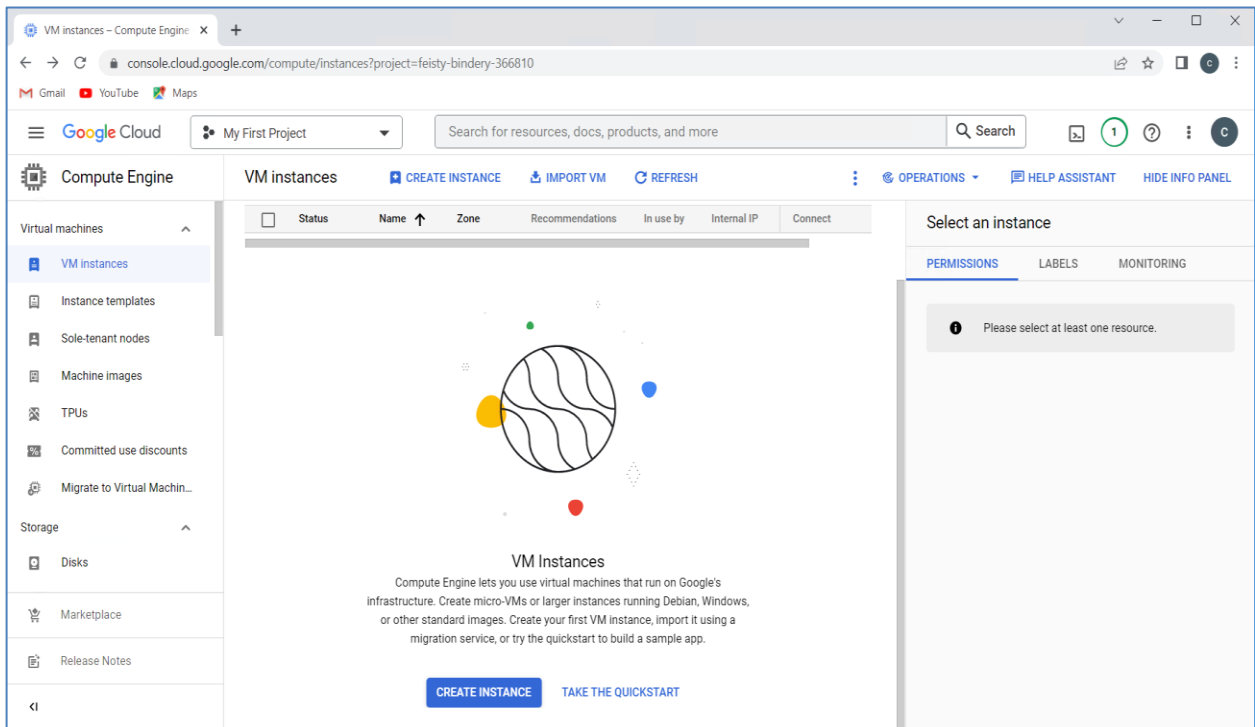
4. Select **Encryption** as **Customer Supplied** and enter the key in text box provided. Copy-Paste the contents of `rsawrapencodedkey.txt` file. Select the **Wrapped key** and click **Create** after providing all the details.



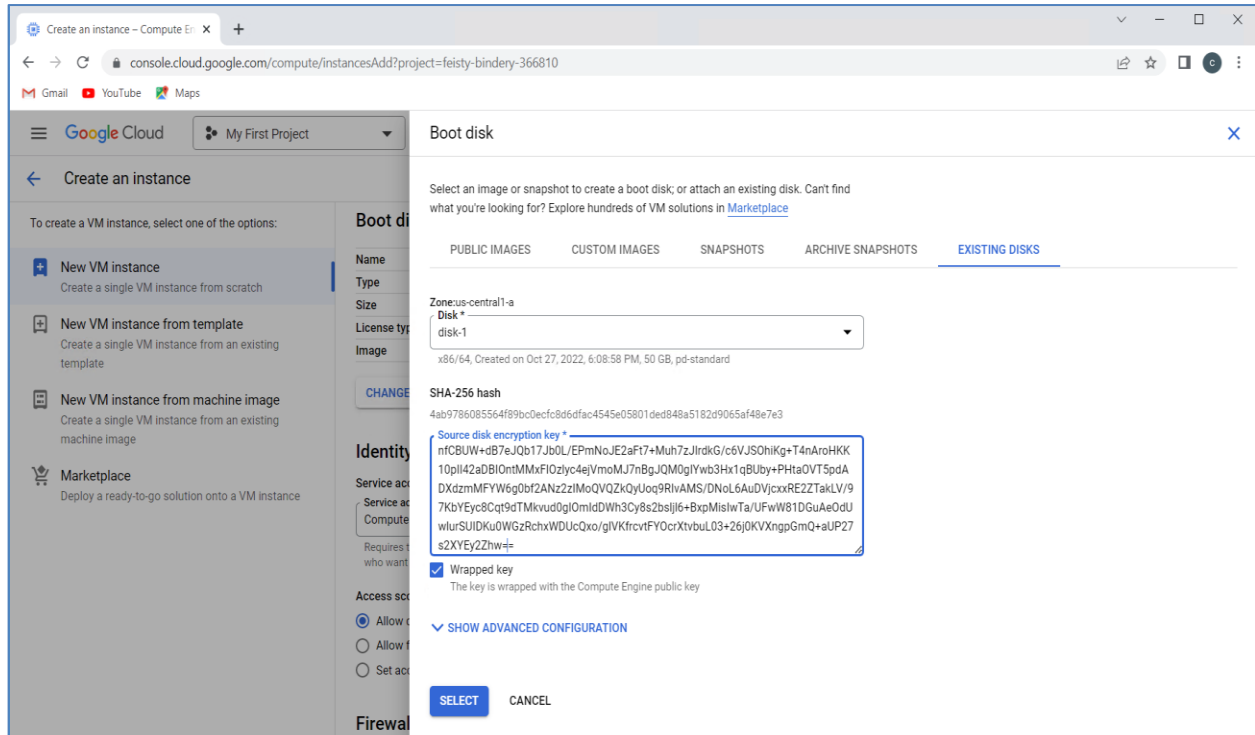
A disk is created, encrypted by wrapped encryption key. You can use this disk to create a VM instance on Google cloud.



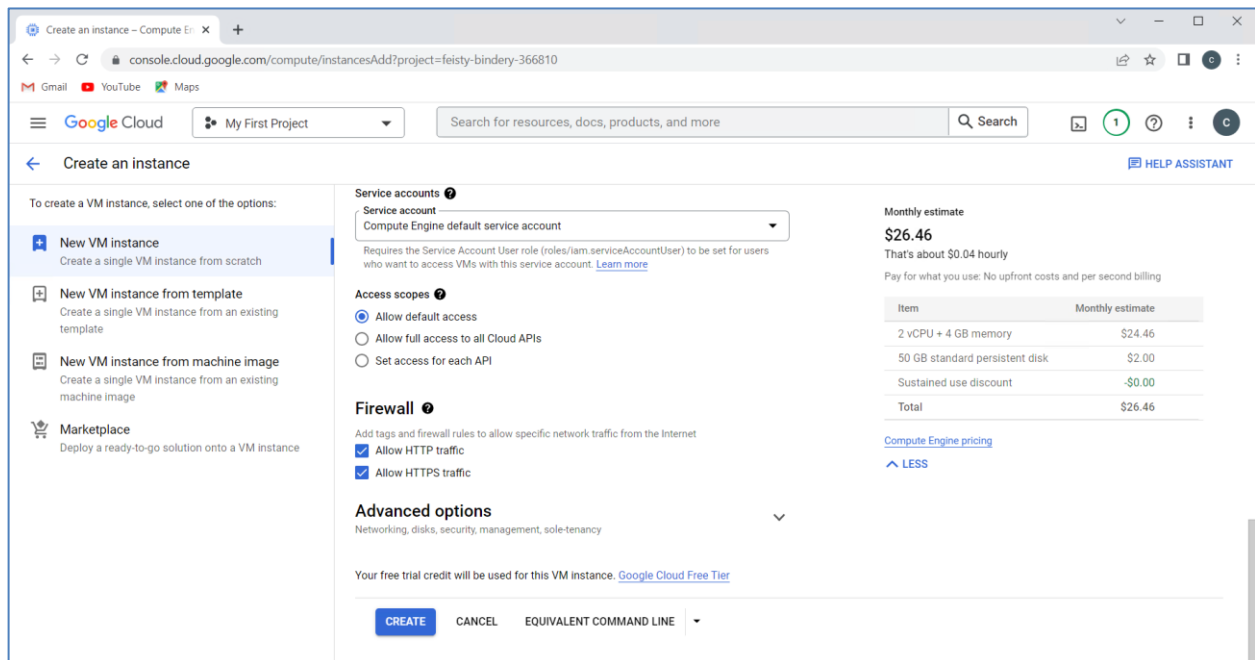
**5. Click VM Instances -> CREATE INSTANCE.**



- Enter the **Name** and select **Zone**, **Machine configuration**. In the **Boot disk** section, click **Change** and then click **Existing disk**. You will see the disk created in the previous steps using CSEK Encryption. When you select the disk, it prompts you to enter the key. Provide the same key that you have used to encrypt the disk and select the **Wrapped key** checkbox. Click **Select**.

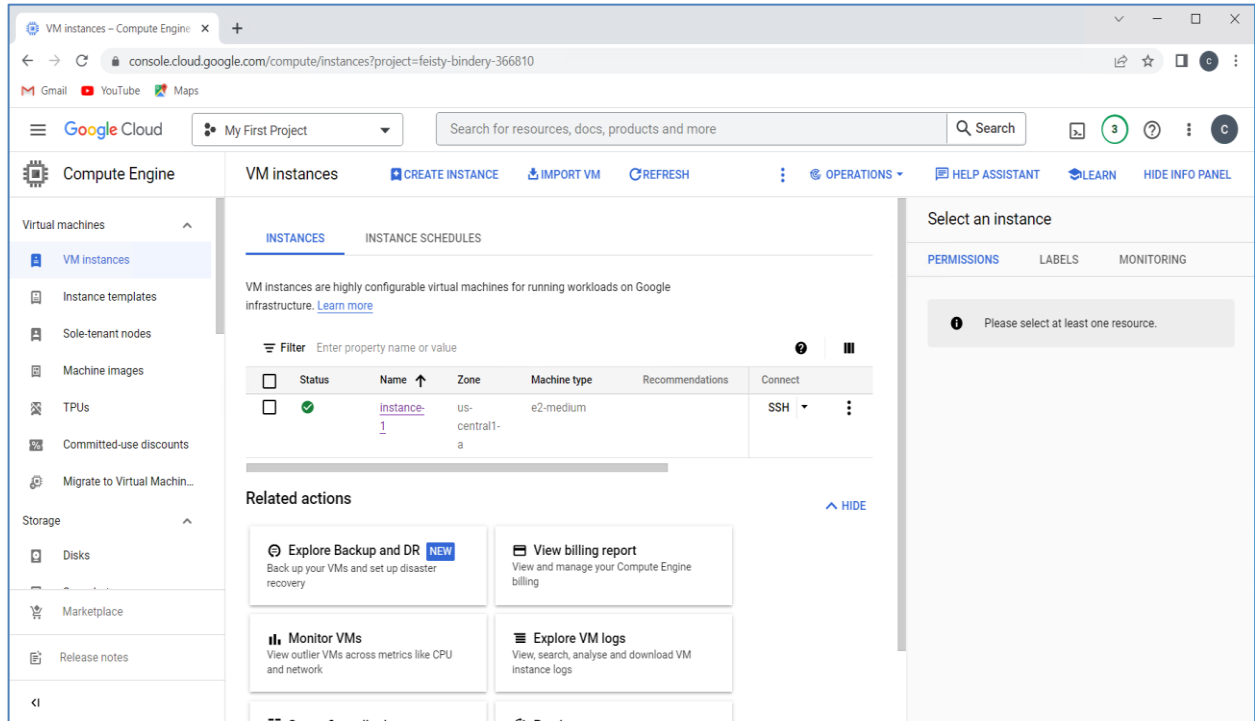


- Select **Allow HTTP traffic** and **Allow HTTPS traffic** in the **Firewall** section and click **Create**.





After a few seconds, your instance will be ready to connect by SSH using the external IP assigned by cloud network.



Refer to Google Cloud Documentation to connect the VM instances through SSH. The steps required for connecting the VM through SSH are provided in the [Appendix](#) as well.

**NOTE:** You cannot start the VM encrypted with CSEK using a cloud console. Use the `gcloud compute` utility, as described next in this Integration Guide to start the VM, or open the cloud shell to start the VM encrypted using CSEK where you need to provide the `csek-key-file` containing the resource encrypted and the wrapped encryption key.

### Using Google Cloud Command-Line Tool

Google Cloud is a part of Google Cloud SDK and it provides various commands to perform operations on Google Cloud. You can use this tool to create encrypted disk or VM using CSEK and start/stop the VM when needed, as well as to perform other operations like creating snapshots from encrypted disk.

When using `gcloud compute` command-line tool to use your keys, you need to provide the encoded keys using a key file that contains your encoded keys as a JSON list. A key file can contain multiple keys, allowing you to manage many keys in a single place. Alternatively, you can create single key files to handle each key separately.

Each entry in your key file must provide:

- > The fully-qualified URI to the resource protected by CSEK
- > The corresponding CSEK
- > The type of key, either `raw` or `rsa-encrypted`

1. Create a file **example-file.json** containing the resource that needs to be encrypted and the encryption key.

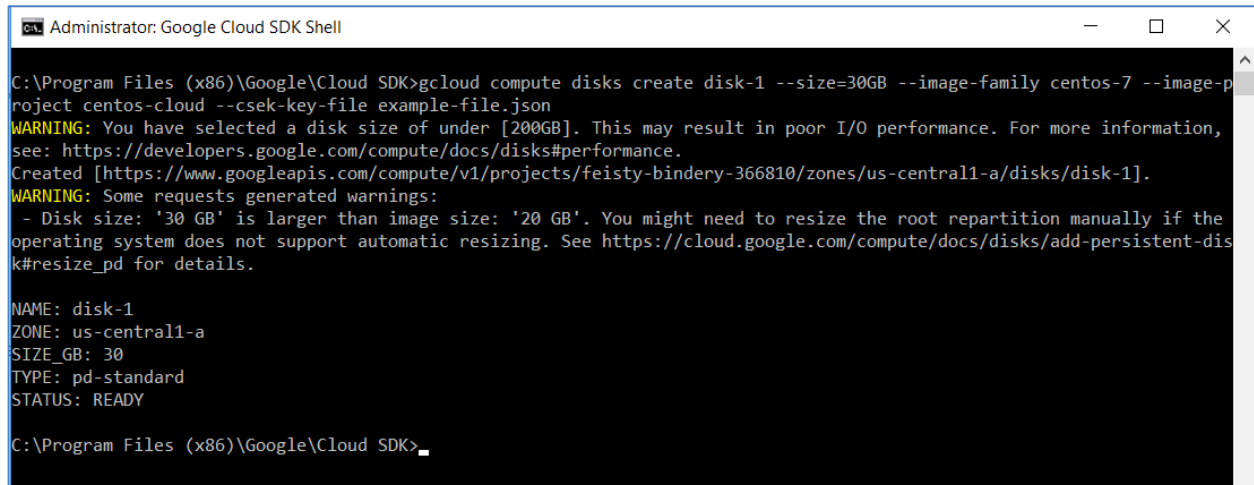
```
[
  {
    "uri": "https://www.googleapis.com/compute/v1/projects/feisty-
bindery-366810/zones/us-central1-a/disks/disk-1",
    "key":
      "MvUdo67dEqy9RbUPxjgXHpcLHAW31MP3XebPdJ9PW0eRnx+zvcVy+3et1j67bSoCphE
wKSFhzVelOB08wMfgeC2cUt2DoIwMKJ3/ZJD3vaQ4nttLMu06/JLFoPAU4kbgJU151Pq
OfaP+t2Ga/mGz7Tj1OSvXesa23wa6oL13ZzAmkGTweVQswJsJQGhm080V0Dc+Wms57qs
s+kobT+KBkMhtV99cqoJHOHoUvywQtXyRcnPmvp2mJRBPE+Qpf07yw09F/4ua2obvvrX
F2GdBY0ym6sZGwu8MHmtHPk1+VJ/ouyIagtO5D+S2eGVWI5aMKpesa1pE2GKmqoq+2er
gTA==",
    "key-type": "rsa-encrypted"
  }
]
```

Here, **uri** refers to the resource that needs to be encrypted using CSEK and **key** refers the base64 encoded RSA wrapped key. Replace “feisty-bindery-366810” and “us-central1-a” with your project and zone, respectively.

**NOTE:** The example provided is for Disks. Similarly, you can create the URI for other resources that need to be encrypted.

2. Create an encrypted disk using CSEK via JSON file.

```
gcloud compute disks create disk-1 --size=30GB --image-family centos-7 --
image-project centos-cloud --csek-key-file example-file.json
```



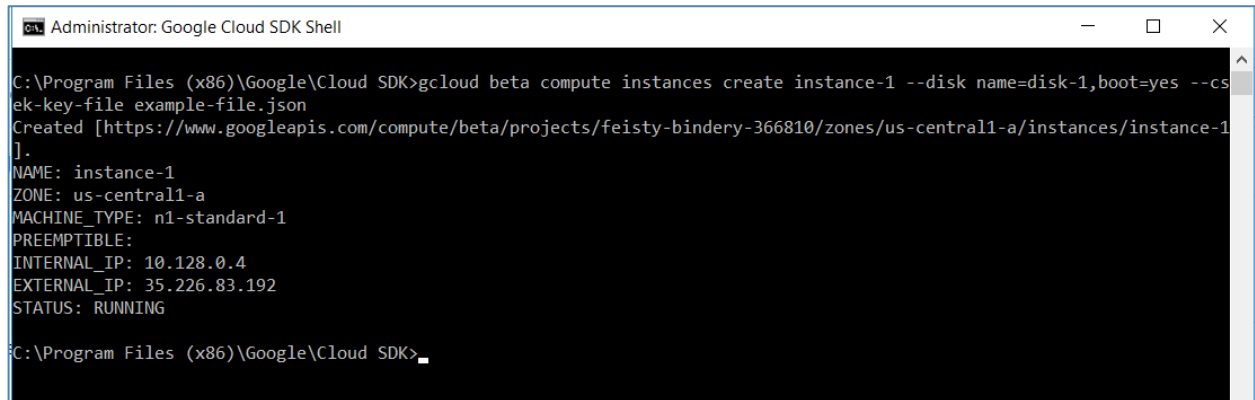
```
Administrator: Google Cloud SDK Shell
C:\Program Files (x86)\Google\Cloud SDK>gcloud compute disks create disk-1 --size=30GB --image-family centos-7 --image-p
project centos-cloud --csek-key-file example-file.json
WARNING: You have selected a disk size of under [200GB]. This may result in poor I/O performance. For more information,
see: https://developers.google.com/compute/docs/disks#performance.
Created [https://www.googleapis.com/compute/v1/projects/feisty-bindery-366810/zones/us-central1-a/disks/disk-1].
WARNING: Some requests generated warnings:
- Disk size: '30 GB' is larger than image size: '20 GB'. You might need to resize the root repartition manually if the
operating system does not support automatic resizing. See https://cloud.google.com/compute/docs/disks/add-persistent-dis
k#resize_pd for details.

NAME: disk-1
ZONE: us-central1-a
SIZE_GB: 30
TYPE: pd-standard
STATUS: READY

C:\Program Files (x86)\Google\Cloud SDK>
```

### 3. Create a VM instance using the encrypted disk.

```
gcloud beta compute instances create instance-1 --disk name=disk-1,boot=yes --csek-key-file example-file.json
```



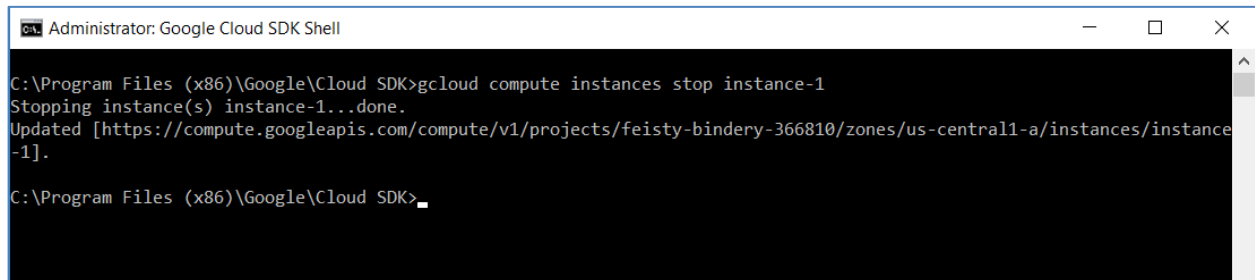
```
Administrator: Google Cloud SDK Shell
C:\Program Files (x86)\Google\Cloud SDK>gcloud beta compute instances create instance-1 --disk name=disk-1,boot=yes --csek-key-file example-file.json
Created [https://www.googleapis.com/compute/beta/projects/feisty-bindery-366810/zones/us-central1-a/instances/instance-1].
NAME: instance-1
ZONE: us-central1-a
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.128.0.4
EXTERNAL_IP: 35.226.83.192
STATUS: RUNNING
C:\Program Files (x86)\Google\Cloud SDK>
```

After creating the VM instance, you can connect your VM through SSH. The steps required for connecting the VM through SSH are provided in the [Appendix](#).

## Stop and Start the VM encrypted by CSEK

### 1. To stop the VM instance, run the following command.

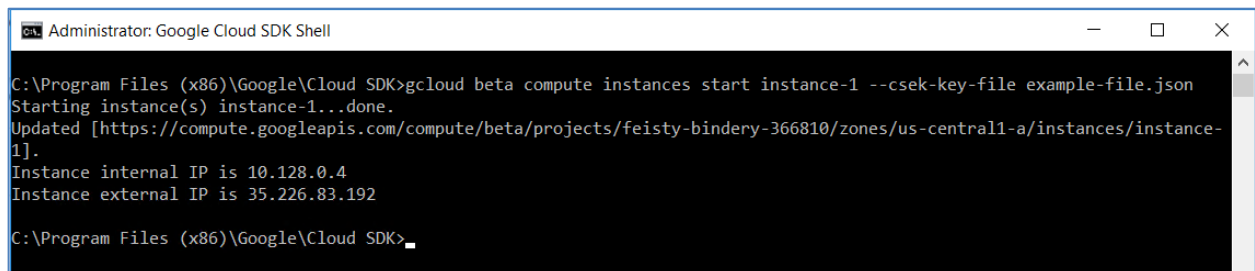
```
gcloud compute instances stop instance-1
```



```
Administrator: Google Cloud SDK Shell
C:\Program Files (x86)\Google\Cloud SDK>gcloud compute instances stop instance-1
Stopping instance(s) instance-1..done.
Updated [https://compute.googleapis.com/compute/v1/projects/feisty-bindery-366810/zones/us-central1-a/instances/instance-1].
C:\Program Files (x86)\Google\Cloud SDK>
```

### 2. To start the VM instance, run the following command.

```
gcloud beta compute instances start instance-1 --csek-key-file example-file.json
```



```
Administrator: Google Cloud SDK Shell
C:\Program Files (x86)\Google\Cloud SDK>gcloud beta compute instances start instance-1 --csek-key-file example-file.json
Starting instance(s) instance-1..done.
Updated [https://compute.googleapis.com/compute/beta/projects/feisty-bindery-366810/zones/us-central1-a/instances/instance-1].
Instance internal IP is 10.128.0.4
Instance external IP is 35.226.83.192
C:\Program Files (x86)\Google\Cloud SDK>
```

Although stopping/deleting the VM instance does not require CSEK, other operations (read/write) such as starting encrypted VM and snapshot of the encrypted disk require CSEK. For details regarding other operations on the encrypted disk, refer to Google Cloud documentation.

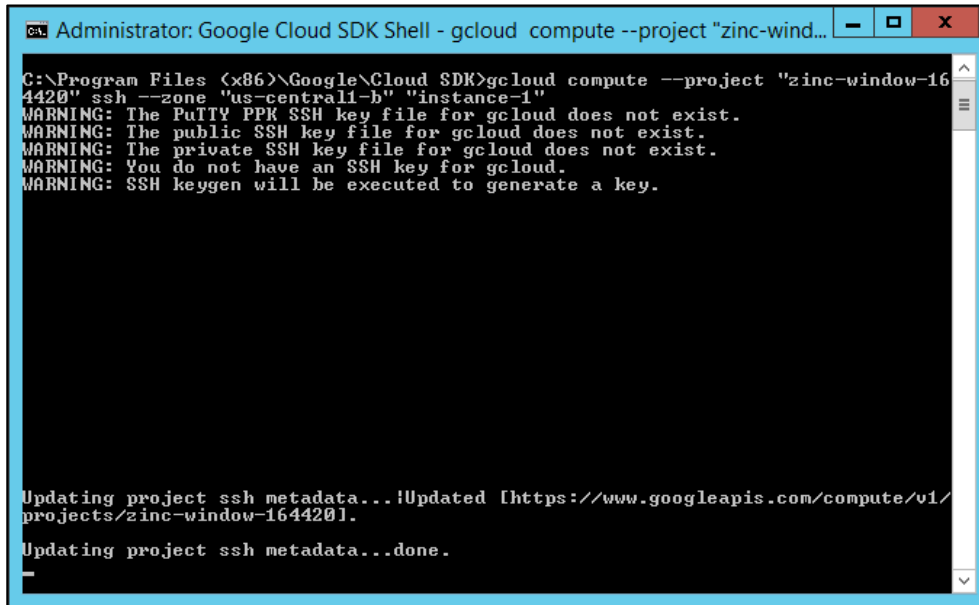
This completes the demonstration of generating the AES256 key on HSM and encrypting the disk using that key on Google Cloud. Each time any read/write operation is performed on the encrypted disk, it prompts for the encryption key and you need to provide the base64 encoded wrapped key. Google keeps the supplied CSEK till the operation is completed. The key is secured on HSM and you can wrap and encode the key when required. If you want to delete the wrapped key from local system, you can delete it. However, there is no harm in keeping the wrapped key as it can be only unwrapped by the Google Private Key.

## Appendix: Connect a VM through SSH

The steps for connecting a VM using SSH are as follows:

1. Open the Google Cloud SDK Shell and run the **gcloud compute** command:

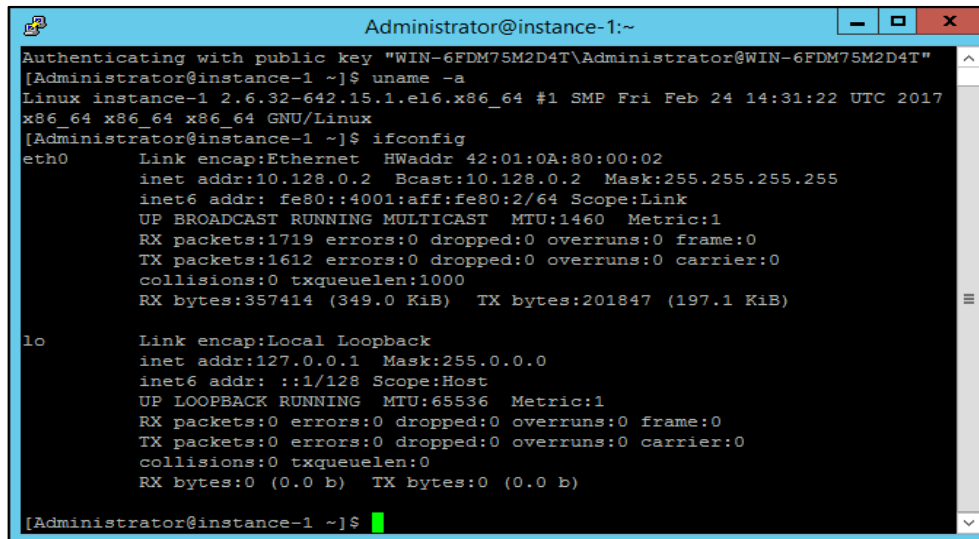
```
# gcloud compute --project "zinc-window-164420" ssh --zone "us-central1-b"
"instance-1"
```



```
C:\Program Files (x86)\Google\Cloud SDK>gcloud compute --project "zinc-window-164420" ssh --zone "us-central1-b" "instance-1"
WARNING: The PuTTY PPK SSH key file for gcloud does not exist.
WARNING: The public SSH key file for gcloud does not exist.
WARNING: The private SSH key file for gcloud does not exist.
WARNING: You do not have an SSH key for gcloud.
WARNING: SSH keygen will be executed to generate a key.

Updating project ssh metadata...Updated [https://www.googleapis.com/compute/v1/projects/zinc-window-164420].
Updating project ssh metadata...done.
```

You'll be connected to an instance using SSH.



```
Administrator@instance-1:~
Authenticating with public key "WIN-6FDM75M2D4T\Administrator@WIN-6FDM75M2D4T"
[Administrator@instance-1 ~]$ uname -a
Linux instance-1 2.6.32-642.15.1.el6.x86_64 #1 SMP Fri Feb 24 14:31:22 UTC 2017
x86_64 x86_64 x86_64 GNU/Linux
[Administrator@instance-1 ~]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 42:01:0A:80:00:02
          inet addr:10.128.0.2  Bcast:10.128.0.2  Mask:255.255.255.255
          inet6 addr: fe80::4001:aff:fe80:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1460  Metric:1
          RX packets:1719 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1612 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:357414 (349.0 KiB)  TX bytes:201847 (197.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[Administrator@instance-1 ~]$
```

When you connect to an instance through the **gcloud** tool, your keys will be generated and applied to your project. These keys will be available at the following locations:

- Public key: **C:\Users\[USER\_NAME]\.ssh\google\_compute\_engine.pub**
- Private key: **C:\Users\[USER\_NAME]\.ssh\google\_compute\_engine**

- To generate a new SSH key-pair on Windows workstations, download putty and puttygen.exe from the following URL:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

**NOTE:** Download 64-bit Windows Installer.

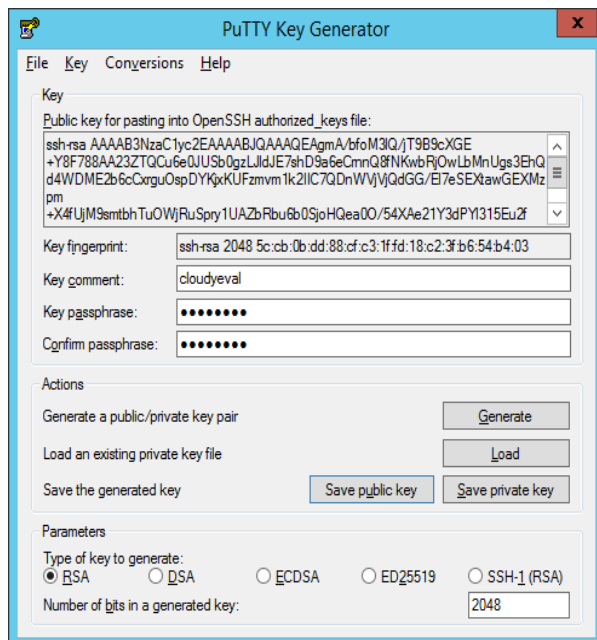
- Run PuTTYgen. For this example, run the puttygen.exe file that you downloaded. A window opens where you can configure your key generation settings.
- Select the default parameters and click **Generate** to generate a new key-pair. When the key generation process gets completed, the tool displays your public key value.
- In the Key comment section, enter your Google username. The key should have the following structure:

```
ssh-rsa [KEY_VALUE] [USERNAME]
```

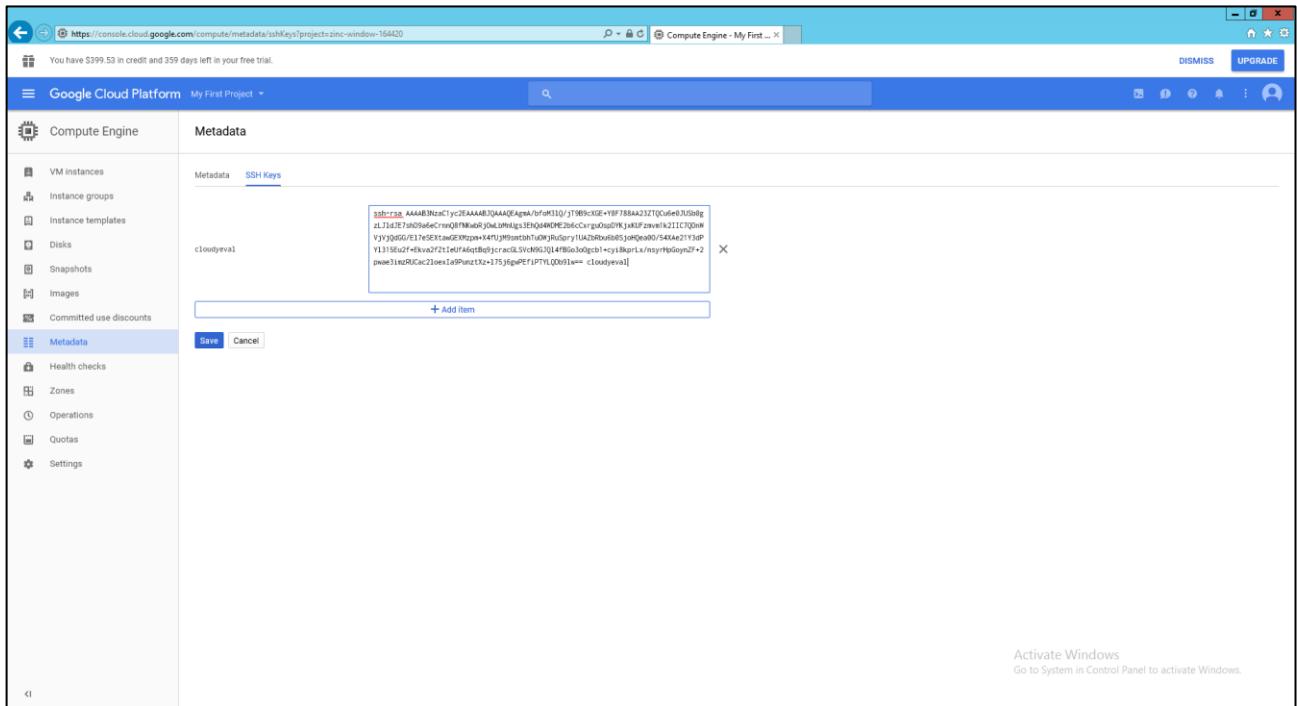
Where:

- [KEY\_VALUE] is the key value that you generated.
- [USERNAME] is your Google username.

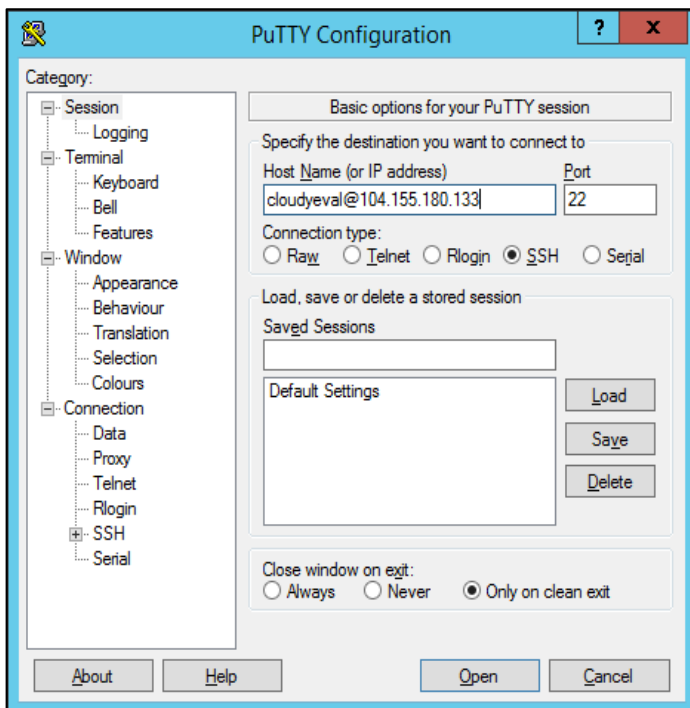
- Optionally, enter a **Key passphrase** to protect your key.
- Click **Save private key** to save the private key to a file. For this example, save the key as **my-ssh-key.ppk**.
- Click **Save public key** to write your public key to a file for later use. Keep the PuTTYgen window open for now.
- In Google Cloud Console, click **Metadata -> SSH Keys -> Edit**.
- Copy the entire public key value from the PuTTYgen tool and paste that value as a new item in the list of **SSH keys** on the **Metadata** page. The public key value is available at the top of the PuTTYgen screen:



- At the bottom of the **SSH Keys** page, click **Save** to save your new project-wide SSH key.

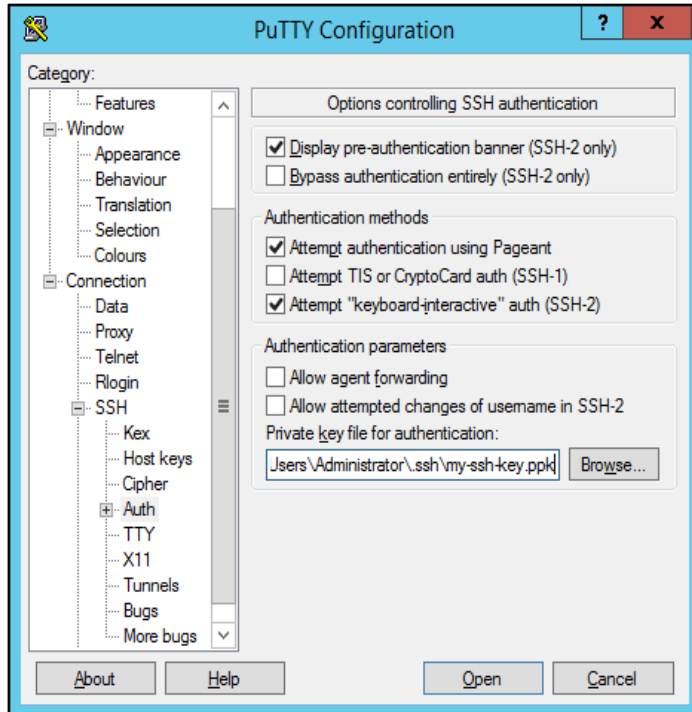


- Run **putty.exe**. In the PuTTY tool, specify your Google username and the external IP address for the instance that you want to connect in the Host Name field. Your username is the Google username that you use to access your project.

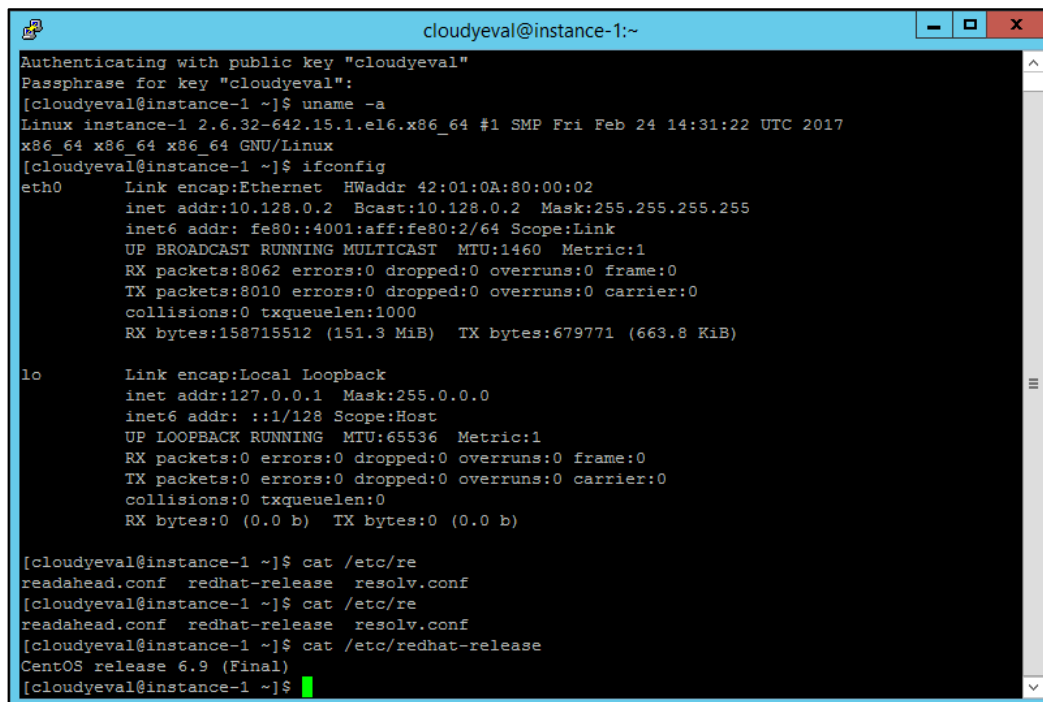


- On the left side of the PuTTY window, navigate to **Connection -> SSH -> Auth.**

- Set the **Private key file for authentication** field with the path to your private key file. For this example, specify the path to the **my-ssh-key.ppk** file.



- Click **Open** to connect with your instance. If the connection is successful, you can use the terminal to run commands on your instance.

The image shows a terminal window titled 'cloudyeval@instance-1:~'. The terminal output is as follows:

```
Authenticating with public key "cloudyeval"  
Passphrase for key "cloudyeval":  
[cloudyeval@instance-1 ~]$ uname -a  
Linux instance-1 2.6.32-642.15.1.el6.x86_64 #1 SMP Fri Feb 24 14:31:22 UTC 2017  
x86_64 x86_64 x86_64 GNU/Linux  
[cloudyeval@instance-1 ~]$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 42:01:0A:80:00:02  
          inet addr:10.128.0.2  Bcast:10.128.0.2  Mask:255.255.255.255  
          inet6 addr: fe80::4001:aff:fe80:2/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1460  Metric:1  
          RX packets:8062 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:8010 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:158715512 (151.3 MiB)  TX bytes:679771 (663.8 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)  
  
[cloudyeval@instance-1 ~]$ cat /etc/re  
readahead.conf  redhat-release  resolv.conf  
[cloudyeval@instance-1 ~]$ cat /etc/re  
readahead.conf  redhat-release  resolv.conf  
[cloudyeval@instance-1 ~]$ cat /etc/redhat-release  
CentOS release 6.9 (Final)  
[cloudyeval@instance-1 ~]$
```



## Contacting Customer Support

---

If you encounter a problem while installing, registering, or operating this product, refer to the documentation. If you cannot resolve the issue, contact your supplier or [Thales Customer Support](#). Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

### Customer Support Portal

The Customer Support Portal, at <https://supportportal.thalesgroup.com>, is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

**NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

### Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.