Google Cloud Platform Customer Supplied Encryption Key (CSEK) Beta

Integration Guide

gemalto
security to be free

Document Number: 007-013795-001, Rev. B

Release Date: August 2017

Contents

Preface

This document is intended to guide administrators through the steps for integrating Google Cloud Platform with a SafeNet Luna HSM to secure CSEK (Customer Supplied Encryption Key) keys. This guide provides the necessary information to install, configure, and integrate Google Cloud Platform with SafeNet Luna HSM.

Scope

This guide provides instructions for setting up a small test lab with Google Cloud Platform running with SafeNet Luna HSM for securing the CSEK keys. It provides information on how to install and configure software that is required for setting up Google Cloud Platform while storing CSEK keys on SafeNet Luna HSM.

NOTE: CSEK feature provided by google cloud is in Beta and this feature is subject to change so customers should proceed with caution when implementing CSEK in production.

Document Conventions

This section provides information on the conventions used in this template.

Notes

Notes are used to alert you to important or helpful information. These elements use the following format:

NOTE: Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. These elements use the following format:

CAUTION: Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. These elements use the following format:

WARNING: Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.

Command Syntax and Typeface Conventions

Convention	Description
bold	The bold attribute is used to indicate the following: • Command-line commands and options (Type **dir /p**.) • Button names (Click **Save As**.) • Check box and radio button names (Select the **Print Duplex** check box.) • Window titles (On the **Protect Document** window, click **Yes**.) • Field names (**User Name:** Enter the name of the user.) • Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) • User input (In the **Date** box, type **April 1**.)
italic	The italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
`Consolas`	Denotes syntax, prompts, and code examples.

Support Contacts

Contact Method	Contact Information	
Address	Gemalto 4690 Millennium Drive Belcamp, Maryland 21017, USA	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Technical Support Customer Portal	https://supportportal.gemalto.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.	

1
Introduction

Overview

This integration guide describes how to store the Customer Supplied Encryption Key (CSEK) on a SafeNet Luna HSM partition. 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 this issue, you can control and manage this encryption yourself, by providing your own encryption keys.

If you provide your own encryption keys, Compute Engine uses your key to encrypt, and therefore protect the Google-generated keys used to encrypt and decrypt your data. Only users who can provide the correct encryption key can use resources protected by a customer-supplied encryption key.

Google does not store your encryption keys on their servers and cannot access your protected data unless you provide the key. 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. In this guide, we will cover the installation and configuration of Google Cloud Platform on Windows Server 2012 R2 using SafeNet Luna HSM.

The benefits of using SafeNet Luna HSM with the Google Cloud Platform are:

- Secure storage of the CSEK Keys
- FIPS 140-2 level 3 validated hardware
- Full life cycle management of the keys

Understanding the Customer Supplied Encryption Key

Server-side encryption refers to encryption that occurs after Cloud Storage receives your data, but before the data is written to disk and stored.

As an alternative to a Google-managed server-side encryption key, you can choose to provide your own AES-256 key, encoded in standard Base64. This key is known as a customer-supplied encryption key (CSEK). 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's 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.

Customer-supplied encryption keys can apply to operations on an object that read or write data. Operations such as deleting or listing objects can be performed without providing the encryption key.

3rd Party Application Details

- Google Cloud Platform

Supported Platforms

- SafeNet Luna HSMs.

Library and Driver Support

- PKCS#11 v2.20 dynamic library

Google Cloud Platform Setup

Google Cloud Platform requires that you have the ability to login in to Google Cloud Console. A Google account is sufficient for login. To use Google Cloud Services, you need to login to Google Cloud Console using your browser and setup your account. The URL for login to Google Cloud Services is provided below:

https://console.cloud.google.com

To begin using the Google Cloud Platform, you need to download and install the Google Cloud SDK on the system you are working on. The Google Cloud SDK provides a set of tools for Cloud Platform. It contains gcloud, gsutil, and bq, which you can use to access Google Compute Engine, Google Cloud Storage, Google BigQuery, and other products and services you can 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 detailed information refer the Google Cloud Online documentation at https://cloud.google.com/docs/.

NOTE: Before proceeding ensure that CSEK feature support is available for your country, if your country is not supported then this feature will not work. List of supported countries for CSEK is available in google cloud online documentation.

https://cloud.google.com/storage/docs/encryption#restrictions

Prerequisites

SafeNet Luna Network HSM Setup

Refer to the SafeNet Luna Network HSM documentation for installation steps and details regarding configuration and setup of the SafeNet Luna Network HSM on Windows/Unix systems. Before you get started ensure that you have a performed the following tasks to prepare the SafeNet Luna Network HSM for use with Google Cloud:

- Configure the SafeNet Luna Network HSM appliance as follows:
  - A secure appliance admin password.
  - A hostname suitable for your network.
  - Network parameters set to work with your network.
- Initialize the HSM on the SafeNet Luna Network HSM appliance to create an HSM SO, cloning domain, and label.
- Create a partition on the HSM and remember the partition password as it will be used by Client to access the partition.
- Use VTL to create, exchange, and register certificates between the SafeNet Luna Network HSM and the Client system to create an NTLS link. Run the `vtl verify` command on the client system to verify the link.
- Enable Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 applies to SafeNet Luna Network HSM with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

2
Integrating Google Cloud Platform with SafeNet Luna HSM

Setting up SafeNet Luna HSM with Google Cloud

HSMs provide strong physical protection of secure assets, including keys, and should be considered a best practice when using cloud.

Before You Begin

Read the VM instances documentation on Google Cloud.

- To use the command-line examples in this guide:
  a. Install the Luna Client and create NTLS with the HSM partition.
  b. Download and install the Open SSL and add openssl.exe location to PATH variable in System Environment.
  c. Install the gcloud command-line tool.
  d. Set a default region and zone.
- Read about disks, images, and persistent disk snapshots.

Generating the CSEK for Google Cloud

After creating the NTLS connection with HSM partition download and import the Google Public Key on the HSM partition which will be use to wrap the generated AES256 key.

To use the CSEK for Google Cloud with SafeNet Luna HSM follow the steps below.

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

   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 SafeNet Luna Client installation directory.

   ```
   # cd “C:\Program Files\SafeNet\LunaClient”
   ```
3. Extract the public key from the certificate using Open SSL:

   ```
   # 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 SafeNet Luna Client.

   ```
   # cmu import -pubkey pubkey.pem -inputFile pubkey.pem -label "google public key"
   ```

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

Provide the partition password when prompted.

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

```
# cmu list
Please enter password for token in slot 0: ********
handle=718      label=google public key
```

Provide the partition password when prompt.

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

```
# C:\Program Files\SafeNet\LunaClient>cmu getAttribute -handle=718
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=a60e0ea3bca01019809738546459b6ef92bdf7d4ea363be08808bfa52cc0252e973b7b1adf8eb36588d9a63e
25e0e3f94f6c6598f5e817f8a06c23bd8c0796f98f0dd5567a2d1bcf43e9dd3f6d99c8bfe488915cd63515ac19bd22dcd319
23b8e19e00efbb8381ad5e01690883ff629a9fad634aa6966867447c28424643535734f122c0e29e8857736cb20c0a68df0a
c0ce77283c70ea40e8d0835f4be62630d67ca0783c149e50dc4c51e787c3d7f5859e03927b1a7336d1af64631aa029c848cb
a6128f277c436d317c672eabae06f600390110b3bbe5d044bf0c3d1d3735689d94e8f7f73ccabd1295c5a0f14cbb5e40f915
0484e40f3ba4e6540c470315
modulusbits=2048
publicexponent=010001
local=false
modifiable=true
```

Where handle is the key handle of the public key. Provide the partition password when prompted.

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

```
# cmu setAttribute -handle=718 -encrypt=true -wrap=true
Please enter password for token in slot 0: ********
```

Where handle is the key handle of the public key. Provide the partition password when prompted.

8. Now create an AES256 key on HSM partition that will be used to encrypt the contents on cloud. To generate the key run the **ckdemo** utility provided with Luna Client.

```
# ckdemo
```

It will show you the available options and prompt for your choice, below are choices (Numeric Values) to generate an AES256 key

```
( 1) Open Session

Enter your choice: 1

( 3) Login

Enter your choice: 3

Crypto Officer    [0]

Crypto User       [1]: 0

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:        715 (0x000002cb)
```

Where 715 is handle of generated AES Key

9. Wrap your key using the public key provided in a certificate that Compute Engine manages. Please ensure to wrap your key using **OAEP** padding. To wrap the key use the same **CKDEMO** session and provide the choices to wrap the AES key using OAEP padding.

```
(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

[30]AES-KW            [35]AES-KEY-WRAP

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): 718

Enter handle of key to wrap (0 to list available objects): 715

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

Where 718 and 715 is the handle of Google Public Key and AES256 key respectively.

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

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

```
Enter your choice: 0

Exiting GESC SIMULATION LAB
```

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

```
# openssl enc -base64 -in wrapped.key > rsawrapencodedkey.txt
```

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

13. Open the Google Cloud SDK Shell and use the **gcloud init** command to perform several common SDK setup tasks. These include authorizing the SDK tools to access Google Cloud Platform using your user account credentials and setting up the default SDK configuration. Installation steps are provided at https://cloud.google.com/sdk/docs/quickstart-windows URL.

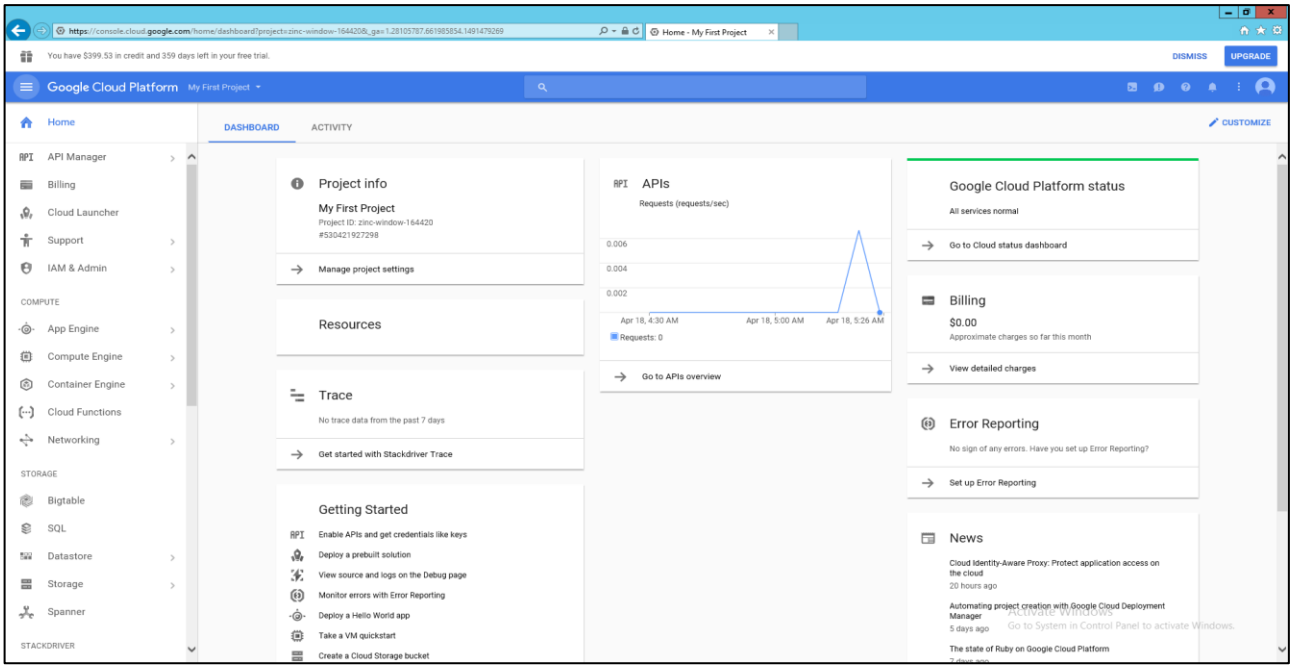
Creating the Encrypted VM using CSEK

Creating an encrypted disk or VM is pretty easy. This guide demonstrated creation of encrypted VM using console and gcloud tool provided by google.

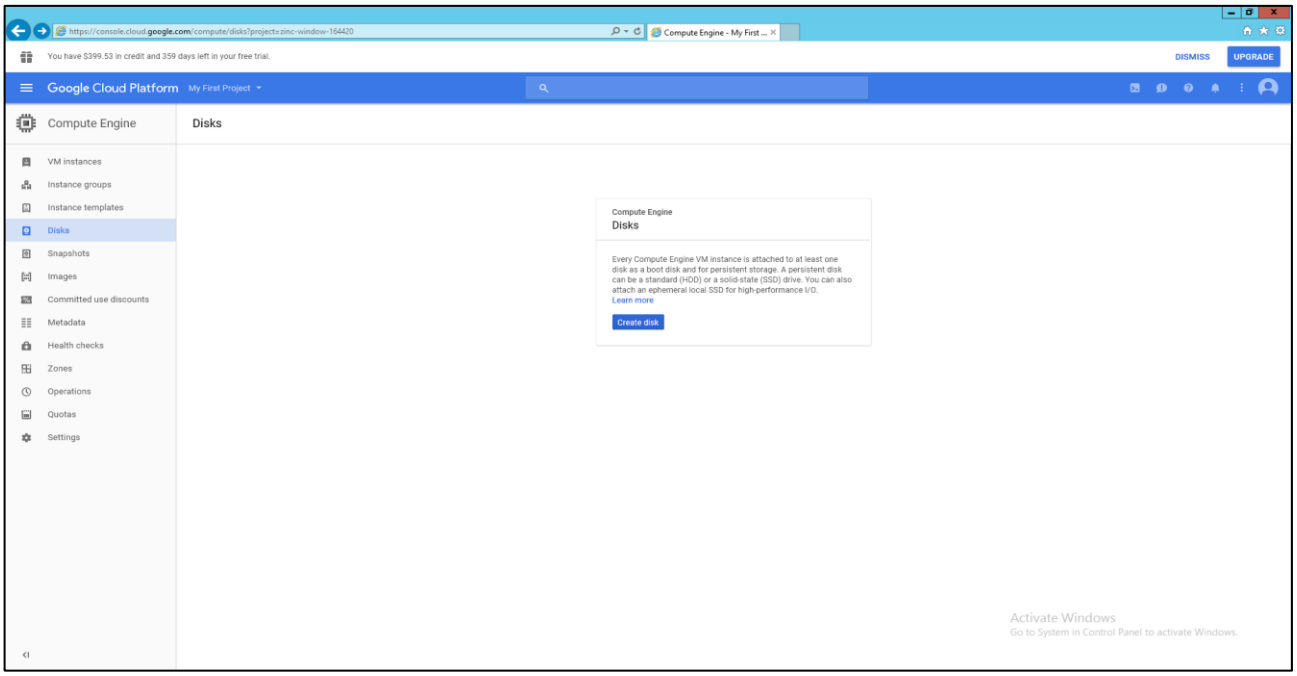
Using Google Console

1. Log on to the Google Cloud Console using the below URL by providing your Google credentials.

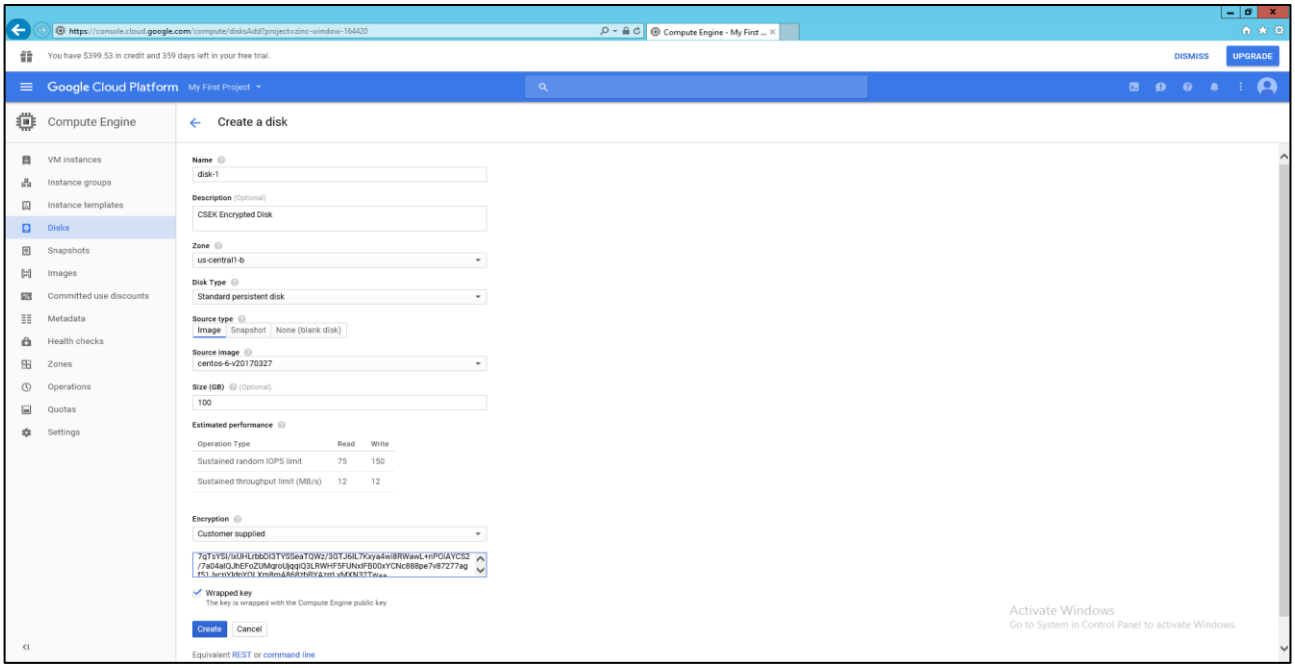
   https://console.cloud.google.com



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

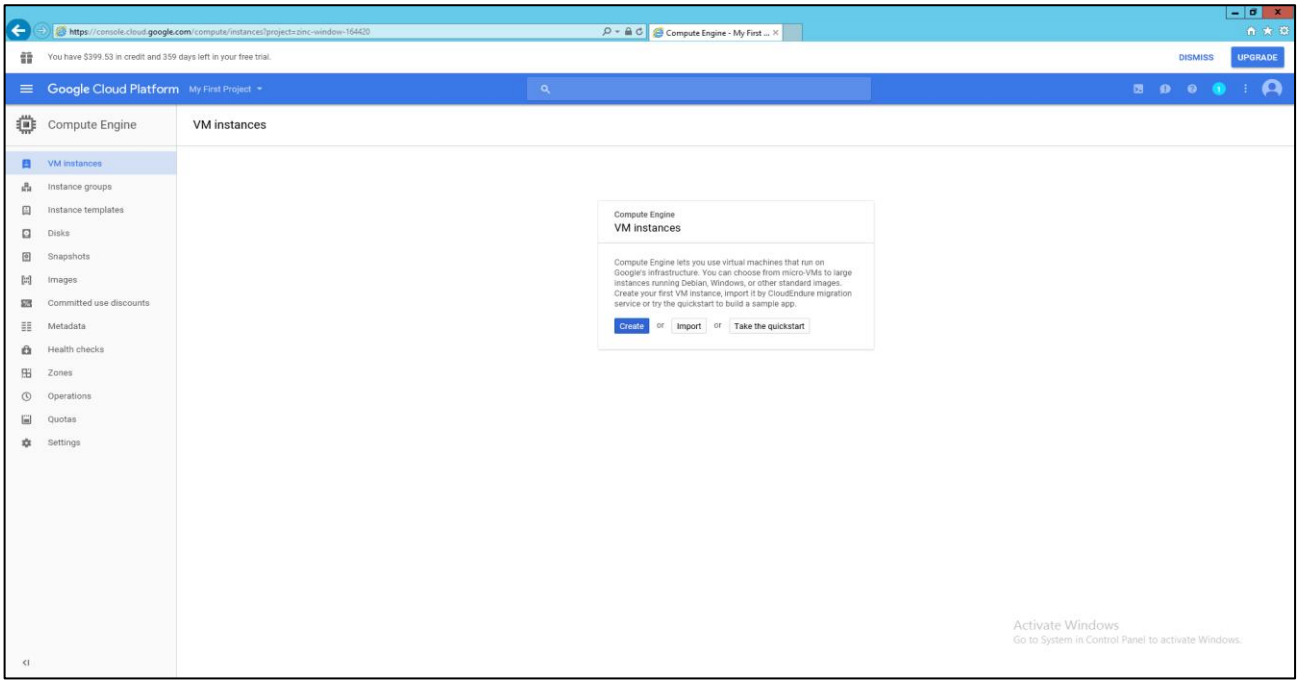


3. Enter the **Name**, **Description**, select **Zone** and **Disk Type** as **Standard persistent disk**. Select **Source type**, **Source Image** (OS that need to be installed) and **Size (GB)**. Select **Encryption** as **Customer Supplied** and enter the key in text box provided. Copy the contents of `rsawrapencodedkey.txt` and paste it. Select the **Wrapped key** and after providing all the details click **Create**.

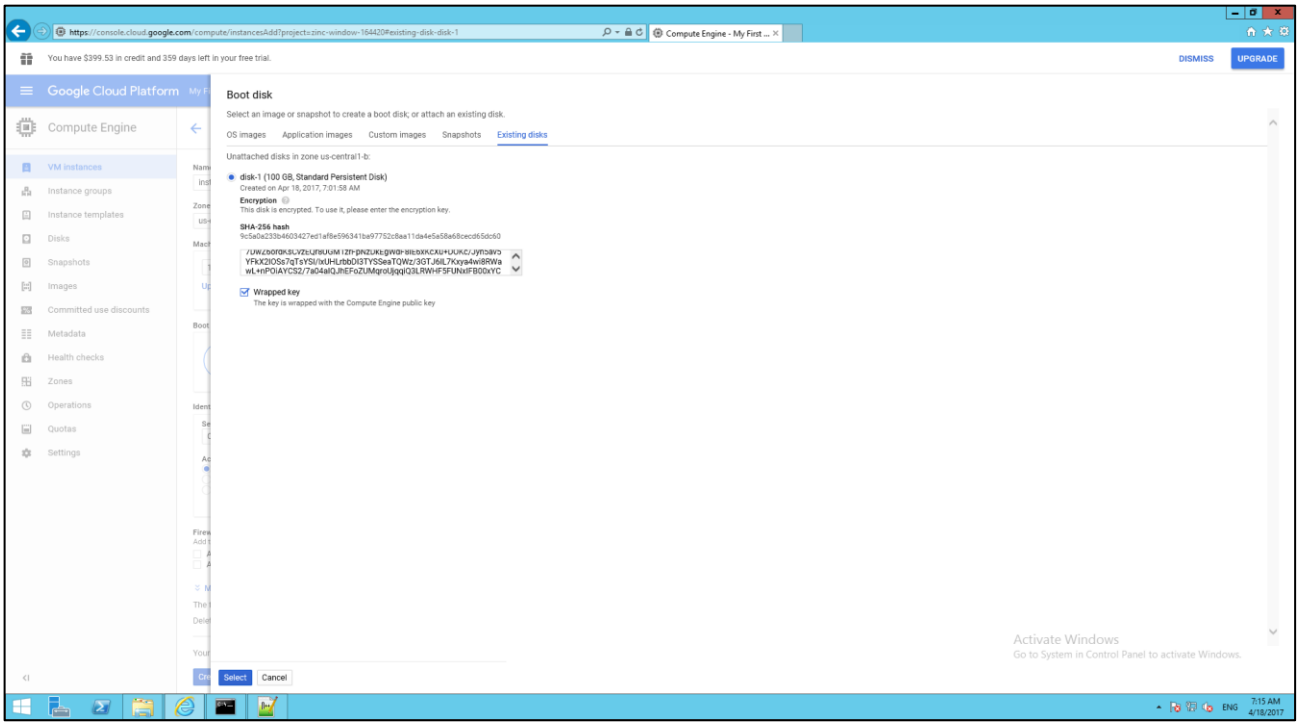


It creates the disk encrypted by customer supplied key and it can be used to create the VM instance on cloud.

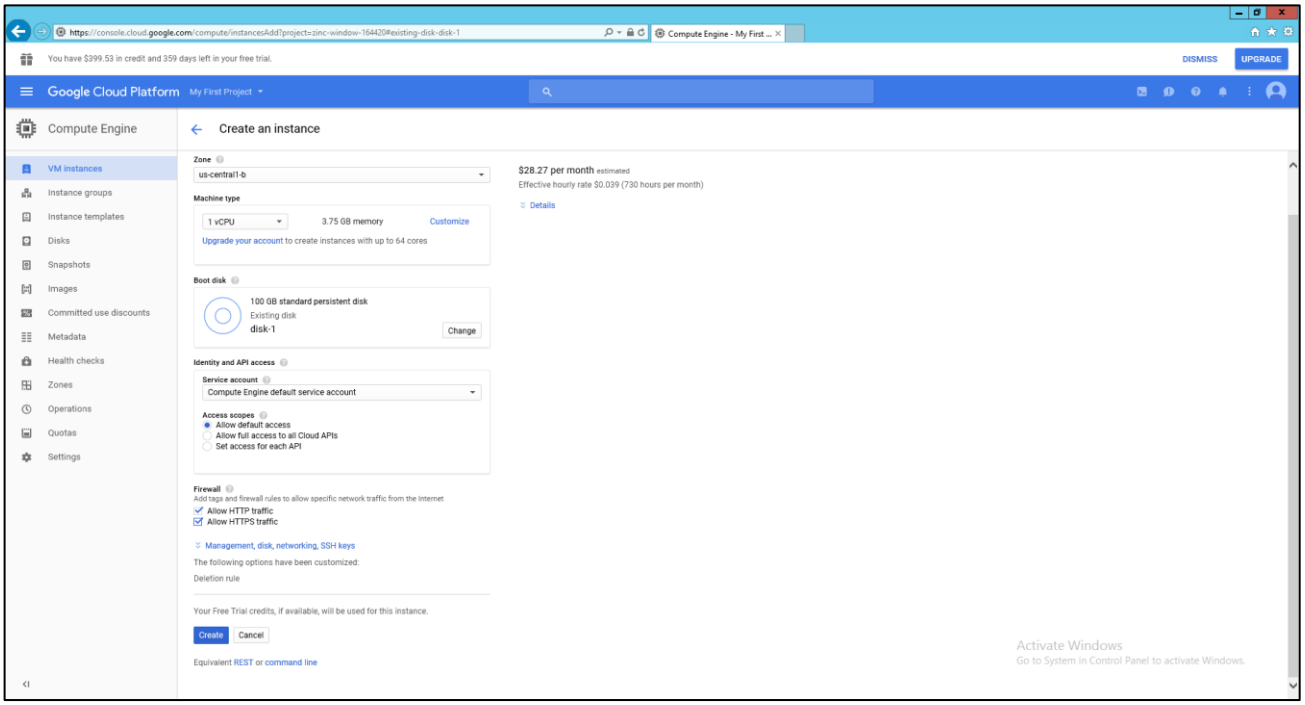
4. Click **VM Instances -> Create**.



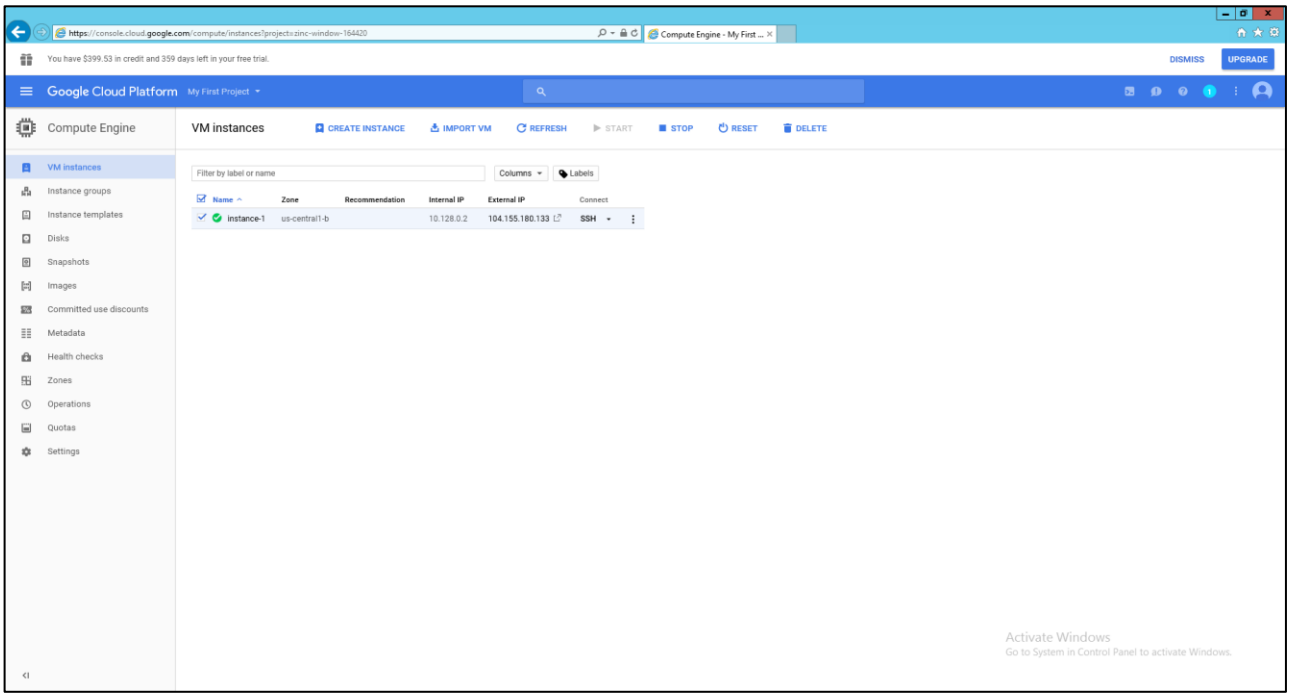
5. Enter the **Name** and select **Zone**, **Machine type**. In the **Boot disk** section, click **Change** and then click **Existing disk**. It displays the disk created in the previous steps using CSEK Encryption. When disk is selected, it prompts to enter the key. Provide the same key which you have used to encrypt the disk and select the **Wrapped key** checkbox. Click **Select**.



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



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



Refer to the Google Cloud Documentation to connect the instances using SSH. Steps for connecting the VM using SSH are provided in the Appendix as an example.

NOTE: The beta version of the CSEK feature includes a console limitation of not being able to start a VM that has been encrypted via CSEK, use the gcloud utility as described next in this Integration Guide to start the VM.

Using gcloud Command-Line Tool

Gcloud is the 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 other operations like creating snapshots from encrypted disk.

1. When you use the gcloud compute command-line tool to set your keys, you provide 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 the key protects
   - The corresponding key
   - The type of key, either raw or rsa-encrypted

   An example key file looks like this:

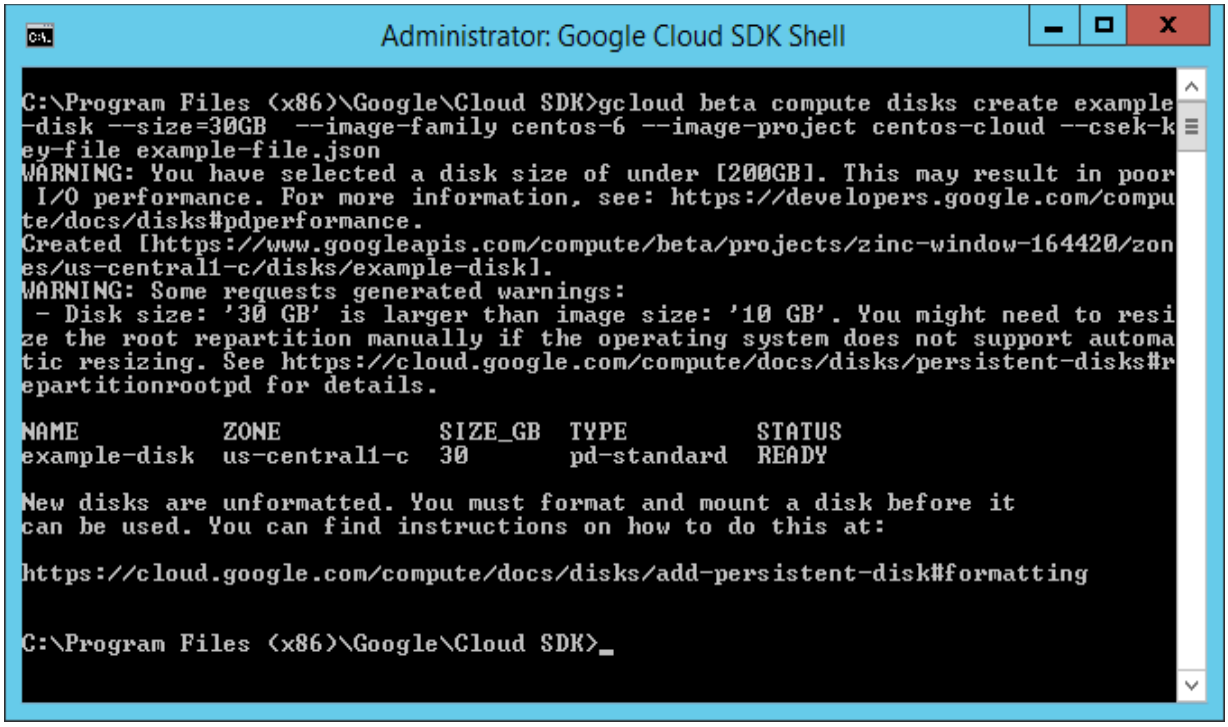
```
[
  {
```

```
"uri": "https://www.googleapis.com/compute/beta/projects/zinc-window-164420/zones/us-central1-c/disks/example-disk",

"key":
"Dj/D5e6cuZmzq+5TPgZolQ+Fji/cnvuaZcvDz0nsxrj/pw/0MViYjo7FbkbIwkyKzpzhgEDZOxNWk7y7rBOQTbYuNR3uxlip/wvuUrYhgZF3BEEEL000aWt67/ZVuFUONJ+hZLQpiQUZ1wKp0nOBdteJtTX7XzEI1OGv+ORv4AGqxEPQGgRHqQB8Jk1afmbGKpw8L1lel0YmkeX5cdjer+5qS2lXdTc0BjdkDF2UsLQYNJS2H3/lIv7/Uk5zH3waKd3YzuQhRt7hEwOM2QS9oE8LiW1v0iaM8Yq2e+XA8MivGNTdra+ZA+29QIVUJ0WZXyNGK8YyxYV5oYNWR+shVQ==",

"key-type": "rsa-encrypted"

  }

]
```


Where example-disk is the name of disk to be created. Replace “zinc-window-164420” and “us-central1-c” with your project and zone respectively.

2. Create an encrypted disk using CSEK supplied by JSON file.

```
# gcloud beta compute disks create example-disk --size=30GB --image-family centos-6 --image-project centos-cloud --csek-key-file example-file.json
```



3. Create a VM instance using the encrypted disk.

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

VM instance is created using encrypted disk now you can connect your VM using SSH using the methods provided in Appendix.

4. You can stop the VM instance using the command below.

```
# gcloud beta compute instances stop example-instance
```

5. To start the VM instance run the following command on gcloud console.

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

Stopping/deleting does not require the CSEK but other operations (read/write) like starting encrypted VM, snapshot of the encrypted disk etc. require the CSEK used to encrypt the disk. For details regarding other operations on encrypted disk refer to the google cloud documentation.

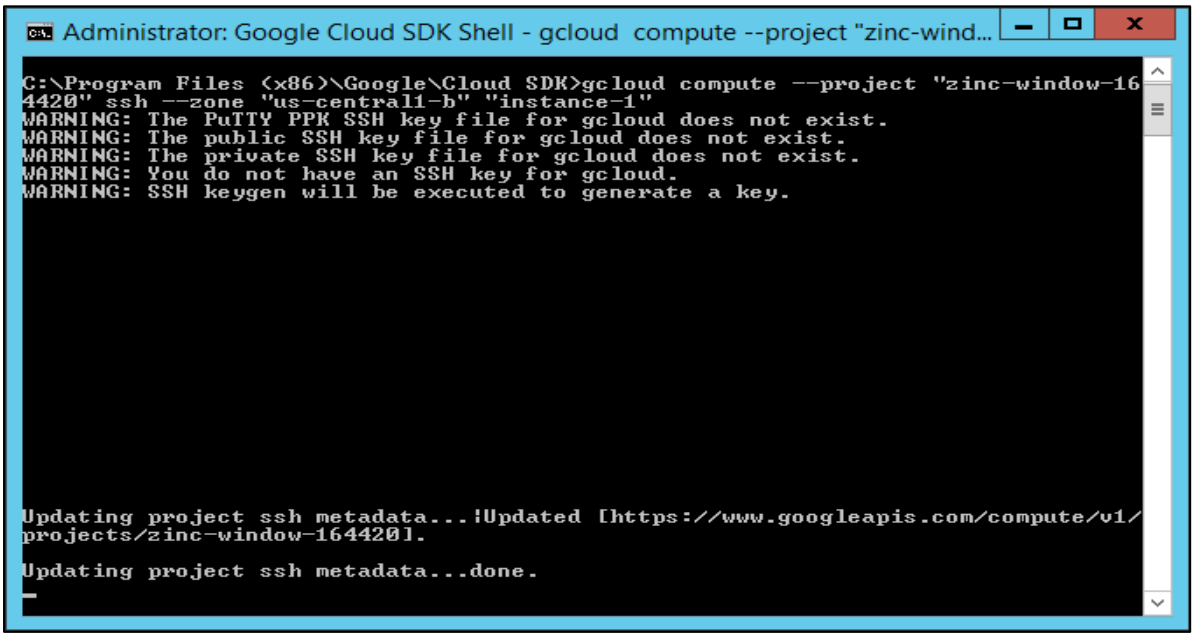
This completes the demonstration of generating the AES256 key on HSM and encrypting the disk using that key on Google Cloud. Each time any read/write operation is performed on encrypted disk, it prompts for the encryption key and you need to provide the base64 encoded wrapped key. Google keep the supplied CSEK till operation completed, for example VM is restarted or snapshot of the encrypted disk, after that the CSEK purge from memory. The key is secured on HSM and you can wrap and encode the key when required. So if you want to delete the wrapped key form local system then 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.

3
Appendix

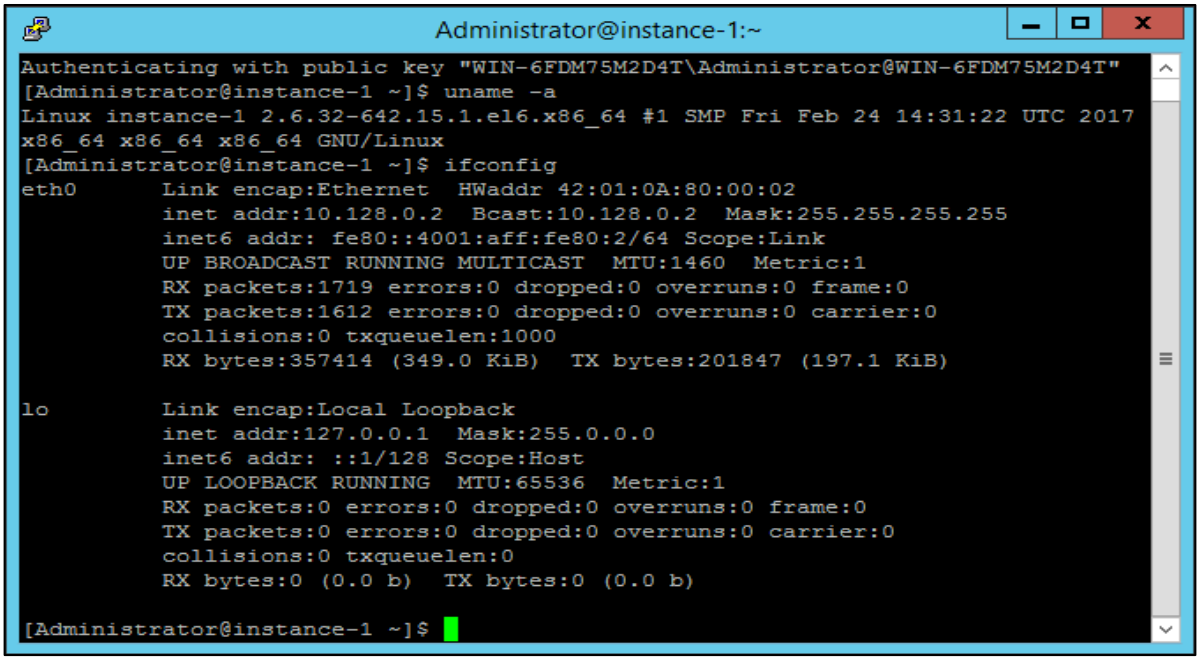
To connect with the VM instances created on google cloud refer the google cloud documentation, however below is the method to connect Linux instance using SSH is provided for your reference.

1. To connect the instances using the gcloud open the Google Cloud SDK Shell and run the **gcloud compute** command as follows:

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

   It connects you to the instance using SSH.



When you connect to an instance through the **gcloud** tool, your keys will be generated and applied to your project and 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**

2. 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

   Download 64 bit Windows Installer.

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

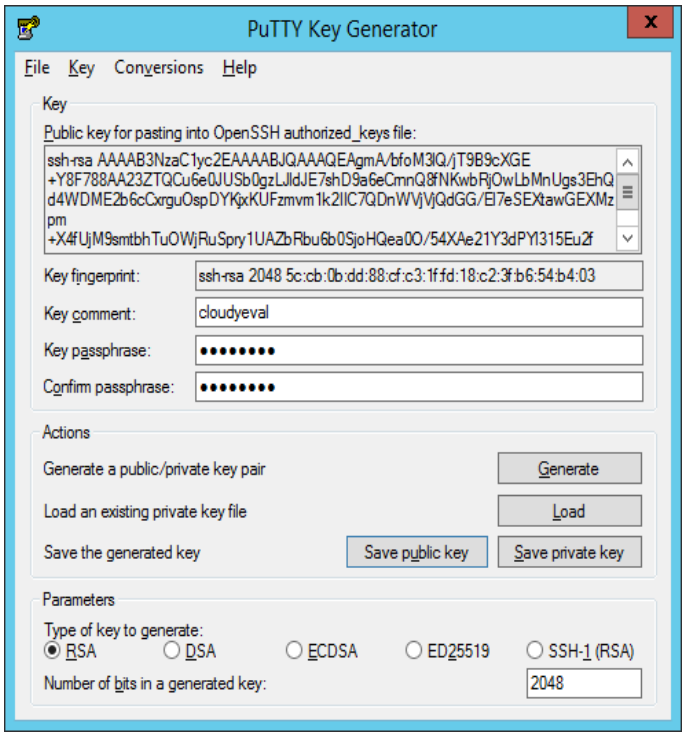
   `ssh-rsa [KEY_VALUE] [USERNAME]`

   `Where:`

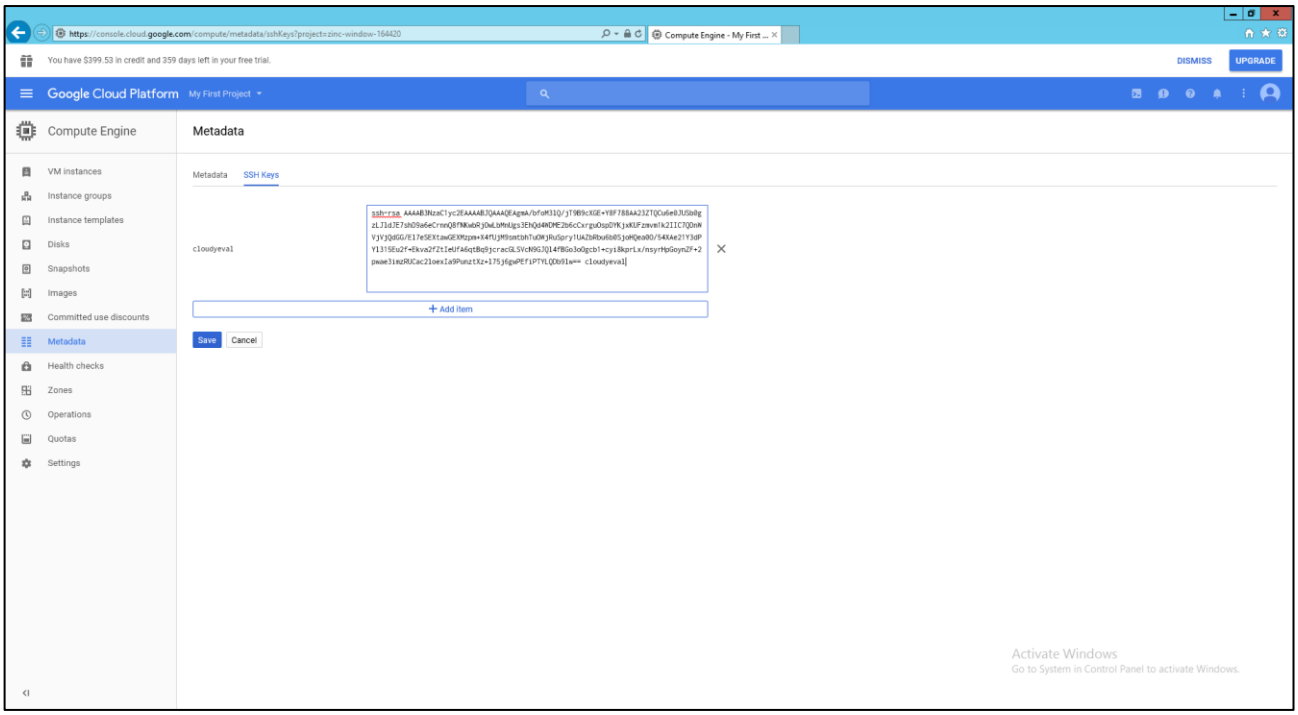
   - `[KEY_VALUE] is the key value that you generated.`
   - `[USERNAME] is your Google username.`

6. Optionally, enter a **Key passphrase** to protect your key.
7. Click **Save private key** to save the private key to a file. For this example, save the key as **my-ssh-key.ppk**.
8. Click **Save public key** to write your public key to a file for use later. Keep the PuTTYgen window open for now.
9. In google cloud console, click **Metadata -> SSH Keys -> Edit**.

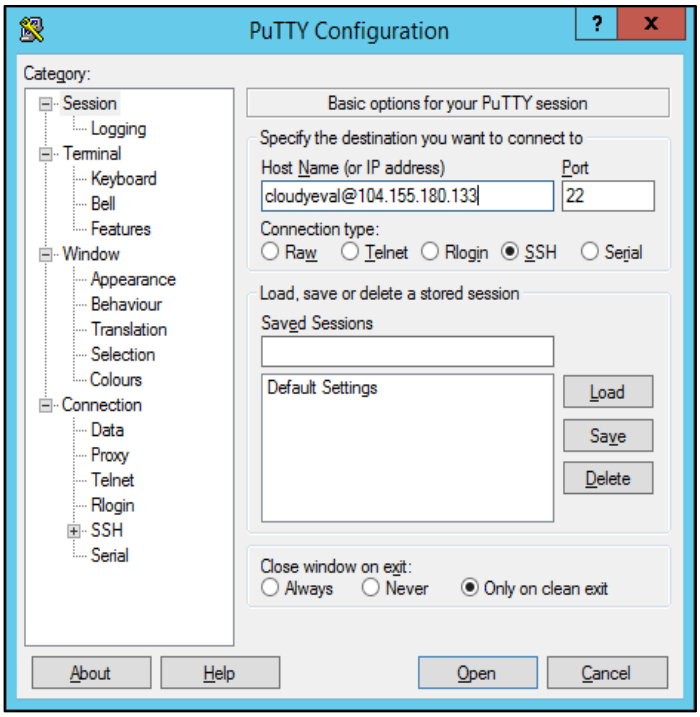
10. Copy the entire public key value from the PuTTYgen tool and paste that value as a new item in the list of **SSH keys** on the **Metadata** page. The public key value is available at the top of the PuTTYgen screen:



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

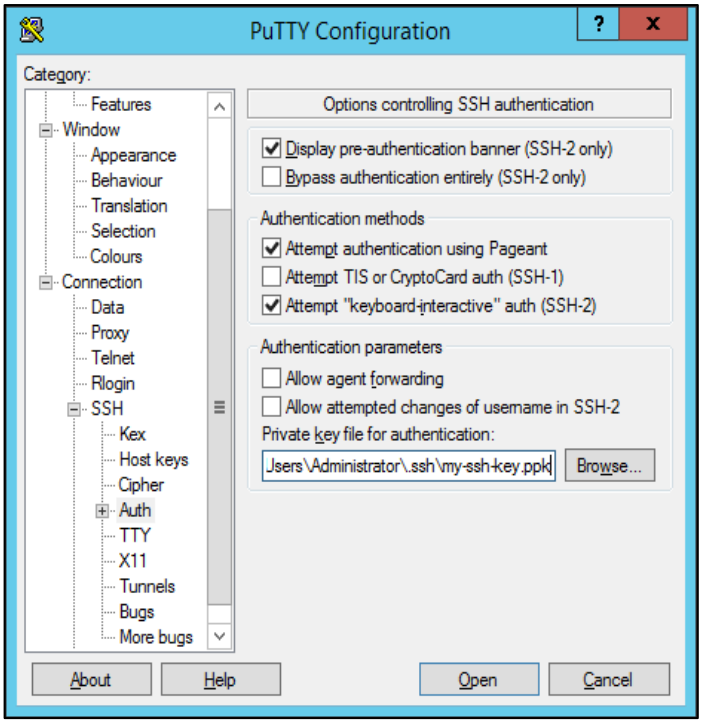


12. Run **putty.exe**. In the PuTTY tool, specify your Google username and the external IP address for the instance that you want to connect in the Host Name field. Your username is the Google username that you use to access your project.



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

14. 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.



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