

Creating a Private Virtual Machine

Overview

Compute Engine lets you create virtual machines that run different operating systems, including multiple flavors of Linux (Debian, Ubuntu, Suse, Red Hat, CoreOS) and Windows Server, on Google infrastructure. You can run thousands of virtual CPUs on a system that is designed to be fast and to offer strong consistency of performance.

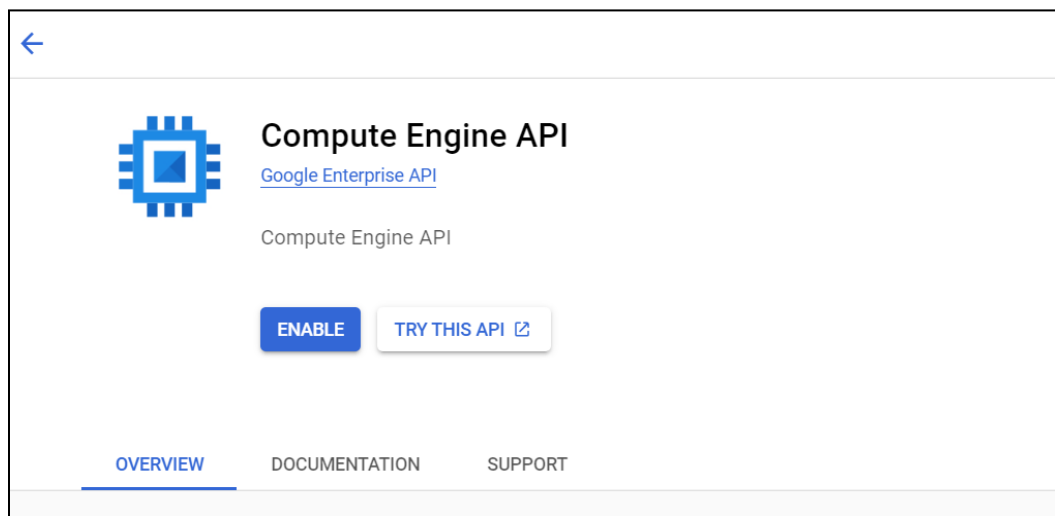
In this hands-on lab, you'll create private virtual machine instances of machine types using the Google Cloud Console.

Prerequisites

Create and manage Google Cloud resources and services directly on the command line.

To create a private virtual machine in GCP following resources such as custom VPC, custom subnets, NAT gateway and IAP firewall rule is required to be provisioned.

Enable the Compute Engine API - [Compute Engine API](#)




1. Create a **Virtual Private Cloud (VPC) & Subnet**

VPC provides networking for your cloud-based resources and services that are global, scalable, and flexible.

Deploying Virtual Private Connect & Subnet using Google Cloud Console

Metadata - VPC & Subnet	Naming Convention
Name	labs-vpc
Subnet Creation Mode	Custom
New Subnet Name Region IP Stack type IPv4 Range Dynamic Routing	labs-subnet us-central1 IPv4 10.0.0.0/24 Regional

- In the Cloud Console, on the **Navigation menu** () , click **VPC network > VPC networks**.
This may take a minute to initialize for the first time.
- To create a new VPC, click **CREATE VPC NETWORK**.

There are many parameters you can configure when creating a **new VPC**.

- Enter VPC **Name** as it is a mandatory field.
- Description can be kept blank as it is an optional field.
- Select **Custom** checkbox for Subnet to create custom subnet.

VPC network

← Create a VPC network

VPC networks

IP addresses

Bring your own IP

Firewall

Routes

VPC network peering

Shared VPC

Serverless VPC access

Packet mirroring

<

Name *

labs-vpc

Lowercase letters, numbers, hyphens allowed

Description

VPC network ULA internal IPv6 range ?

Enabling this feature will assign a /48 from Google defined ULA prefix fd20::/20.

☐ Enabled
 ☒ Disabled

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)

Subnet creation mode ?

☒ Custom
 ☐ Automatic

- Click on **Add Subnet**.
- Enter Subnet **Name** as it is a mandatory field.
- Description can be kept blank as it is an optional field.
- Region here selected is us-central1. Region can be chosen upon end user's requirements to obtain low latency of resources.
- For more information about regions, see the Compute Engine guide, [Regions and Zones](#).
- IP Stack type can be left default with IPv4 checkbox selected.
- Enter **IPv4 range** for Subnet depending upon requirement.
- Rest of the fields can be left with default values.
- Click the **CREATE** button to create a new VPC.

New subnet ^

Name * ?

Lowercase letters, numbers, hyphens allowed

Description

Region * ▼ ?

IP stack type

☒ IPv4 (single-stack)

☐ IPv4 and IPv6 (dual-stack) ?

IPv4 range * ?

E.g. 10.0.0.0/24

[CREATE SECONDARY IPV4 RANGE](#)

Dynamic routing mode ?

☒ Regional
Cloud Routers will learn routes only in the region in which they were created

☐ Global
Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router

DNS server policy ▼ ?

Maximum transmission unit (MTU) ▼

[CREATE](#) [CANCEL](#)

EQUIVALENT COMMAND LINE ▼

- Following resources are created
- VPC named **labs-vpc** and subnet named **labs-subnet**.

▼ labs-vpc	1	1460	Custom	None
us-central1	labs-subnet	10.0.0.0/24	None	

2. Create **NAT Gateway**

Cloud NAT ([network address translation](#)) lets certain resources without external IP addresses create outbound connections to the internet.

Deploying NAT Gateway using Google Cloud Console

Metadata - NAT Gateway	Naming Convention
Name	labs-nat-gw
Select Cloud Router Network Region	labs-vpc us-central1
Create a Router Name	labs-nat-router
Cloud NAT Mapping Resource (Internal)	Primary and Secondary Ranges for all subnets

- In the Cloud Console, on the **Navigation menu** (≡), click **Network services > Cloud NAT**.
- To create a new NAT gateway, click **CREATE CLOUD NAT GATEWAY**.

There are many parameters you can configure when creating a new NAT gateway

- Enter NAT **Gateway name** as it is a mandatory field.
- In Select Cloud Router, select the VPC Network created previously i.e. labs-vpc
- Select the region associated with the subnet.

Network services

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

Marketplace

Release Notes

Create Cloud NAT gateway

Cloud NAT lets your VM instances and container pods communicate with the internet using a shared, public IP address.

Cloud NAT uses Cloud NAT gateway to manage those connections. Cloud NAT gateway is region and VPC network specific. If you have VM instances in multiple regions, you'll need to create a Cloud NAT gateway for each region. [Learn more](#)

Gateway name *

labs-nat-gw

Lowercase letters, numbers, hyphens allowed

Select Cloud Router

Network *

labs-vpc

Region *

us-central1 (Iowa)

One subnet.

Cloud Router *

- In Cloud Router, click on create new router.

Network services

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

Marketplace

Release Notes

Create Cloud NAT gateway

Region *

us-central1 (Iowa)

One subnet.

Cloud Router *

Filter |Type to filter

Create new router

Primary and secondary ranges for all subnets

Cloud NAT IP addresses

Automatic (recommended)

Destination (external)

Internet

ADVANCED CONFIGURATIONS

- Enter the Cloud Router name as it is a mandatory field. Rest of the fields can be left with default values.
- Click on **CREATE** to create a Cloud router.

Create a router

Google Cloud Router dynamically exchanges routes between your Virtual Private Cloud (VPC) and on-premises networks by using Border Gateway Protocol (BGP)

Name *
labs-nat-router ?
Lowercase letters, numbers, hyphens allowed

Description

Network *
labs-vpc ?

Region *
us-central1 (Iowa) ?

BGP peer keepalive interval seconds ?

CREATE CANCEL

- The router's name is populated in the Cloud Router field.
- Rest of the fields can be left with default values.

Create Cloud NAT gateway

us-central1 (Iowa) ?

One subnet.

Cloud Router *
labs-nat-router ?

Cloud NAT mapping ?

Source (internal) ?
Primary and secondary ranges for all subnets ?
Select which subnets to map to the Cloud NAT gateway. Primary IP addresses are used by VM instances and secondary IP addresses are used by container pods. [Learn more](#)

Cloud NAT IP addresses
Automatic (recommended) ?

Destination (external)
Internet

ADVANCED CONFIGURATIONS

CREATE CANCEL

- Cloud NAT gateway named **labs-nat-gw** is created.

<input type="checkbox"/>	labs-nat-gw	us-central1	labs-nat-router	✓ Running	⋮
--------------------------	-----------------------------	-------------	---------------------------------	-----------	---


3. Create IAP Firewall Rule.

To allow IAP to connect to your VM instances, create a firewall rule that:

- applies to all VM instances that you want to be accessible by using IAP.
- allows ingress traffic from the IP range **35.235.240.0/20**. This range contains all IP addresses that IAP uses for TCP forwarding.
- allows connections to all ports that you want to be accessible by using IAP TCP forwarding, for example, port 22 for SSH and port 3389 for RDP.

Deploying IAP Firewall using Google Cloud Console

Metadata - Firewall	Naming Convention
Name	labs-iap-fw-rule
Network	labs-vpc
Direction of Traffic Action on Match	Ingress Allow
Target	All Instance in Network
Source Filter	IPv4 Ranges
Source IPv4 Ranges	35.235.240.0/20
Protocol & Ports TCP	Specified Protocols & Ports 22, 3389

- In the Cloud Console, on the **Navigation menu** () , click **VPC network > Firewall**.
- To create a new IAP Firewall rule, click **CREATE FIREWALL RULE**.

There are many parameters you can configure when creating a new firewall rule

- Enter firewall **Name** as it is a mandatory field.
- Description can be left blank as it is an optional field.
- Select the VPC **Network** created previously i.e. **labs-vpc** to attach the firewall to that network.

- Click on **CREATE** to create an IAP firewall.

- The IAP firewall named **labs-iap-fw-rule** is created.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs	Hit count
labs-iap-fw-rule	Ingress	Apply to all	IP ranges: 35.235.240.0/20	tcp:22, 3389	Allow	1000	labs-vmc	Off	

4. Create Service Account for Compute Engine

A service account is identified by its email address, which is unique to the account. Before creating the service account following the below steps

- Enable the IAM API - [IAM API](#)

Required roles for your IAM account.

To get the permissions that you need to manage service accounts, grant the following IAM roles on the project:

To view and create service accounts:

Create Service Accounts (roles/iam.serviceAccountCreator)

Deploying Custom Service Account using Google Cloud Console

Metadata - Service Account	Naming Convention
Service Account Details Service Account Name	labs-compute-sa
Add Principals New Principals	labs-compute-sa
Assign roles	Storage Admin

- In the Cloud Console, on the **Navigation menu** (≡), click **IAM & Admin > Service Account**.
- To create a new custom service account, click **CREATE SERVICE ACCOUNT**.

There are many parameters you can configure when creating a new firewall rule

- Enter **Service Account Name** as it is a mandatory field.
- Service account ID is auto populated with service account name.
- Description can be kept blank as it is an optional field.
- Click **CREATE AND CONTINUE** to create the custom service account.

The screenshot displays the 'Create service account' interface in the Google Cloud Console. On the left, the 'IAM & Admin' navigation menu is visible, with 'Service Accounts' highlighted. The main content area is titled 'Create service account' and shows the following details:

- Step 1: Service account details**
 - Service account name:** labs-compute-sa (Display name for this service account)
 - Service account ID:** labs-compute-sa
 - Email address:** labs-compute-sa@[redacted].iam.gserviceaccount.com
 - Service account description:** (Optional field)
 - CREATE AND CONTINUE** button
- Step 2: Grant this service account access to project (optional)**
- Step 3: Grant users access to this service account (optional)**

- Other fields are optional and can be skipped. The custom service account name **labs-compute-sa@project-id.iam.gserviceaccount.com** is created.

Filter labs-compute-sa Enter property name or value						
<input type="checkbox"/>	Email	Status	Name ↑	Key ID	Key creation date	Actions
<input type="checkbox"/>	labs-compute-sa@[REDACTED].iam.gserviceaccount.com	✓	labs-compute-sa	No keys		⋮

- Navigate to IAM, click **Grant ACCESS**.
- Enter the created service account name in **New Principals**.
- In Role, Select **Storage Admin** from the dropdown.


Setup for Private Compute Engine

Create Compute Engine

Enable the Compute Engine API - [Compute Engine API](#)

Deploying Google Compute Engine (GCE) using Google Cloud Console

Metadata - VM Instance	Naming Convention
Name	labs-demo-vm
Region	us-central1
Zone	us-central1-a
Series Machine Type	E2 e2-medium
Boot Disk Operating System Version Size Boot Disk Type	Ubuntu Ubuntu 18.04 LTS 10 GB Balanced Persistent Disk
Service Account	labs-compute-sa
Network Interface Network Sub Network External IPv4 address	labs-vpc labs-subnet None
Automation Startup Script	<pre>#!/bin/bash apt update apt -y install apache2 cat <<EOF > /var/www/html/index.html <html><body><p>Linux startup script added directly.</p></body></html></pre>

- In the Cloud Console, on the **Navigation menu** () , click **Compute Engine > VM Instances**.
- To create a new instance, click **CREATE INSTANCE**.
- Enter the **Name** of VM instance.
- Select region and zone to create VM in. Here region is selected as us-central1 and zone is us-central1-a
- In Machine Configuration, there are various Machines types available [About machine families](#). For this lab, E2 series medium machine type is selected which is a 2-CPU, 4GB RAM instance

← Create an instance

To create a VM instance, select one of the options:

New VM instance

Create a single VM instance from scratch

New VM instance from template

Create a single VM instance from an existing template

New VM instance from machine image

Create a single VM instance from an existing machine image

Marketplace

Deploy a ready-to-go solution onto a VM instance

Name *

labs-demo-vm

Labels

+ ADD LABELS

Region *

us-central1 (Iowa)

Region is permanent

Zone *

us-central1-a

Zone is permanent

Machine configuration

Machine family

GENERAL-PURPOSE COMPUTE-OPTIMIZED MEMORY-OPTIMIZED GPU

Machine types for common workloads, optimized for cost and flexibility

Series

E2

CPU platform selection based on availability

Machine type

e2-medium (2 vCPU, 4 GB memory)

vCPU

1-2 vCPU (1 shared core)

Memory

4 GB

✓ CPU PLATFORM AND GPU

Monthly estimate

\$25.46

That's about \$0.03 hourly

Pay for what you use: No upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Sustained use discount	-\$0.00
Total	\$25.46

[Compute Engine pricing](#)

[^ LESS](#)

- Select the Boot disk Image, by default debian image is selected. Click on change to select different types of OS image.

← Create an instance

To create a VM instance, select one of the options:

New VM instance

Create a single VM instance from scratch

New VM instance from template

Create a single VM instance from an existing template

New VM instance from machine image

Create a single VM instance from an existing machine image

Marketplace

Deploy a ready-to-go solution onto a VM instance

Display device

Enable to use screen capturing and recording tools.

☐ Enable display device

Confidential VM service

Confidential Computing is disabled on this VM instance

ENABLE

Container

Deploy a container image to this VM instance

DEPLOY CONTAINER

Boot disk

Name	labs-demo-vm
Type	New balanced persistent disk
Size	10 GB
License type	Free
Image	Ubuntu 18.04 LTS

CHANGE

Monthly estimate

\$25.46

That's about \$0.03 hourly

Pay for what you use: No upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Sustained use discount	-\$0.00
Total	\$25.46

[Compute Engine pricing](#)

[^ LESS](#)

- A Pop-up will appear to select the required Operating System and Version. Several images are available, including Debian, Ubuntu, CoreOS, and premium images such as Red Hat Enterprise Linux and Windows Server. Rest fields can be kept with default values.
- Select **Ubuntu** in the Operating system and version **Ubuntu 18.04 LTS**.
- Select **Balanced persistent disk** Boot disk type and Size **10 GB**.

The screenshot shows the 'Create an instance' page in Google Cloud. On the left, there's a sidebar with options: 'New VM instance' (Create a single VM instance from scratch), 'New VM instance from template' (Create a single VM instance from an existing template), 'New VM instance from machine image' (Create a single VM instance from an existing machine image), and 'Marketplace' (Deploy a ready-to-go solution onto a VM instance). The main area is titled 'Boot disk' and contains the following fields:

- Operating system:** A dropdown menu with 'Ubuntu' selected.
- Version:** A dropdown menu with 'Ubuntu 18.04 LTS' selected. Below it, a note says 'x86_64, amd64 bionic image built on 2022-10-18, supports Shielded VM features'.
- Boot disk type:** A dropdown menu with 'Balanced persistent disk' selected.
- Size (GB):** A text input field with '10' entered.
- Buttons:** 'COMPARE DISK TYPES' (link), 'SHOW ADVANCED CONFIGURATION' (toggle), 'SELECT' (button), and 'CANCEL' (button).

- In the identity and API access section, select the service account created previously named **labs-compute-sa** from the service account dropdown.
- In Firewall, do not select any checkbox as this will be private virtual. If there is a requirement, then firewall rules can be created. Click on Advanced options.

The screenshot shows the 'Identity and API access' section of the 'Create an instance' page. It includes the following fields and sections:

- Service accounts:** A dropdown menu with 'labs-compute-sa' selected. Below it, a note says 'Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. [Learn more](#)'.
- Access scopes:** A section with a note 'Use IAM roles with service accounts to control VM access. [Learn more](#)'.
- Firewall:** A section with a note 'Add tags and firewall rules to allow specific network traffic from the Internet'. It contains two checkboxes: 'Allow HTTP traffic' and 'Allow HTTPS traffic', both of which are unchecked.
- Advanced options:** A section with a chevron icon to its right.
- Networking:** A section with a chevron icon to its right and a subtitle 'Hostname and network interfaces'. It contains two text input fields: 'Network tags' and 'Hostname', both with a question mark icon to their right. Below the 'Hostname' field, a note says 'Set a custom hostname for this instance or leave it default. Choice is permanent'.

- In Advanced options, **Networking** is an important field to decide where this virtual image resides.

In the **Edit network interface** field, select the previously created network i.e labs-vpc and select the subnetwork (subnet).

In **External IPv4 Address**, select **None** to disable external IP to virtual machine.

The screenshot shows the 'Create an instance' wizard in Google Cloud Platform, specifically the 'Edit network interface' step. On the left, there are three options to create a VM instance: 'New VM instance' (selected), 'New VM instance from template', and 'New VM instance from machine image'. Below these is the 'Marketplace' option. The main area is titled 'Edit network interface' and contains several settings: 'Networks in this project' is selected; 'Network' is set to 'labs-vpc'; 'Subnetwork' is set to 'labs-subnet IPv4 (10.0.0.0/24)'; 'IP stack type' is set to 'IPv4 (single-stack)'; 'Primary internal IP' is set to 'Ephemeral (Automatic)'; 'Alias IP ranges' has an 'ADD IP RANGE' button; and 'External IPv4 address' is set to 'None'. On the right, a 'Monthly estimate' section shows a total of \$25.46, broken down into '2 vCPU + 4 GB memory' (\$24.46), '10 GB balanced persistent disk' (\$1.00), and a 'Sustained use discount' of -\$0.00. A 'DONE' button is at the bottom right.

- (This Field is Optional) Expand Management, In **Automation** Startup Scripts can be added which will perform action post creation of VM. A startup script is a file that contains commands that run when a virtual machine (VM) instance boots. Compute Engine provides support for running startup scripts on Linux VMs and Windows VMs. Please refer to the below screenshot.

Use the below Automation Script

```
#!/bin/bash
apt update
apt -y install apache2
cat <<EOF > /var/www/html/index.html
<html><body><p>Linux startup script added directly.</p></body></html>
```

- Rest of the fields can be left with default values. Click on **CREATE** to create the private virtual image.

To create a VM instance, select one of the options:

- New VM instance**
Create a single VM instance from scratch
- New VM instance from template**
Create a single VM instance from an existing template
- New VM instance from machine image**
Create a single VM instance from an existing machine image
- Marketplace**
Deploy a ready-to-go solution onto a VM instance

Disks
Additional disks

Security
Shielded VM and SSH keys

Management
Description, deletion protection, reservations, automation, and availability policies

Description

Deletion protection ⓘ
☐ Enable deletion protection

Reservations
Application policy
Automatically use created reservation
Use an existing reservation when creating this VM instance

Automation
Startup script
#!/bin/bash
apt update
apt -y install apache2
cat <<EOF > /var/www/html/index.html
<html><body><p>Linux startup script added directly.</p></body></html>

- On the console, check the status of the newly created virtual machine.

Filter **lab-demo** Enter property name or value

<input type="checkbox"/>	Status	Name ↑	Zone	Machine type	Recommendations	Internal IP	External IP	Labels	Connect
<input type="checkbox"/>	✓	lab-demo-vm	us-central1-a	e2-medium		10.0.0.3 (nic0)			SSH ▾

This instance is running

Related actions

HIDE

Connect to the virtual machine

- To securely access the private virtual machine, Following IAP Permission/roles is required to user
 - IAP-secured Tunnel User (roles/iap.tunnelResourceAccessor)**
 - The permissions can be set in **IAM & Admin** for each individual user.
- Click on SSH dropdown, various connect options are available. Preferred and mostly used is **Open in the browser window**.

- Check the connectivity to the internet by pinging to google.com.

```
$ ping google.com
```

```
root@lab-demo-vm:~# ping google.com
PING google.com (173.194.195.100) 56(84) bytes of data.
64 bytes from iw-in-f100.1e100.net (173.194.195.100): icmp_seq=1 ttl=115 time=1.23 ms
64 bytes from iw-in-f100.1e100.net (173.194.195.100): icmp_seq=2 ttl=115 time=0.526 ms
64 bytes from iw-in-f100.1e100.net (173.194.195.100): icmp_seq=3 ttl=115 time=0.376 ms
64 bytes from iw-in-f100.1e100.net (173.194.195.100): icmp_seq=4 ttl=115 time=0.430 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3041ms
rtt min/avg/max/mdev = 0.376/0.641/1.233/0.346 ms
root@lab-demo-vm:~#
```

- Verify the Startup Script. The script installs the apache2 package and overrides the index file. Curl localhost to get the output.

```
$sudo systemctl status apache2
```

```
root@labs-demo-vm-3:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2022-11-09 12:32:21 UTC; 4min 14s ago
     Main PID: 3489 (apache2)
        Tasks: 55 (limit: 4660)
   CGroup: /system.slice/apache2.service
           └─3489 /usr/sbin/apache2 -k start
             └─3490 /usr/sbin/apache2 -k start
               └─3491 /usr/sbin/apache2 -k start

Nov 09 12:32:21 labs-demo-vm-3 systemd[1]: Starting The Apache HTTP Server...
Nov 09 12:32:21 labs-demo-vm-3 systemd[1]: Started The Apache HTTP Server.
root@labs-demo-vm-3:~# curl localhost
<html><body><p>Linux startup script added directly.</p></body></html>
root@labs-demo-vm-3:~#
```

Install Packages

- Install C++ compiler package g++ using the following command:-

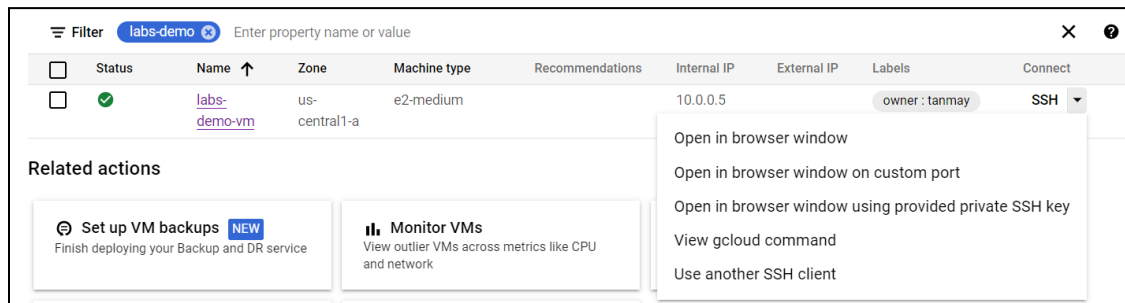
```
sudo apt update
sudo apt install g++
g++ --version
```

```
root@lab-demo-vm:~# g++ --version
g++ (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

root@lab-demo-vm:~#
```

Connecting Virtual Machine using GCP Cloud Shell Editor

- Click on SSH dropdown, various connect options are available. Select **View gcloud command**.




- Gcloud command line Pop-up appears with gcloud command to connect to VM. Click on **RUN IN CLOUD SHELL**.

gcloud command line

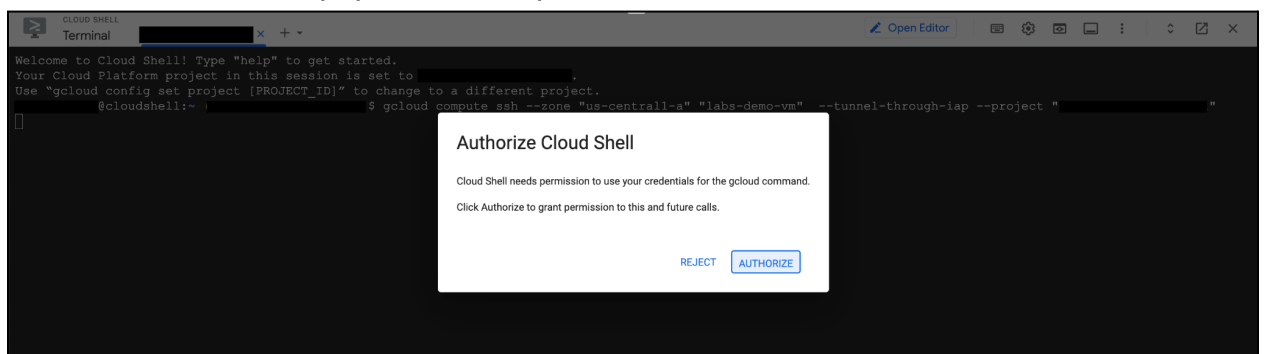
The following gcloud command line can be used to SSH into this instance. [gcloud reference](#)

```
$ gcloud compute ssh --zone "us-central1-a" "labs-demo-vm" --tunnel-through-iap --project "██████████"
```

☒ Line wrapping

 COPY TO CLIPBOARD RUN IN CLOUD SHELL CLOSE

- Post clicking on RUN IN CLOUD SHELL, the cloud shell editor window will open with the gcloud command. Click on **ENTER**.
- Authorize Cloud Shell pop will show up. Click on **AUTHORIZE**.



- Congratulations you have successfully logged into VM using cloud shell editor.

```
@cloudshell:~ $ gcloud compute ssh --zone "us-central1-a" "labs-demo-vm" --tunnel-through-iap --project "
WARNING:
To increase the performance of the tunnel, consider installing NumPy. For instructions,
please see https://cloud.google.com/iap/docs/using-tcp-forwarding#increasing_the_tcp_upload_bandwidth
Warning: Permanently added 'compute.8857660453860978307' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1098-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Jan 16 09:06:35 UTC 2023

System load:  0.0           Processes:      105
Usage of /:   18.7% of 9.51GB Users logged in:   0
Memory usage: 5%           IP address for ens4: 10.2.0.19
Swap usage:   0%

0 updates can be applied immediately.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@labs-demo-vm:~$
```

References

- Virtual Private Cloud overview - [VPC](#)