

Virtual Machine Instance

In Google Cloud

Stamatis Sideris

DATA MANAGEMENT & ANALYTICS CONSULTANT

Contents

Introduction	1
What is a Virtual Machine Instance (VM Instance).....	1
VM Instance Installation using Google Cloud	2
Creation of Google Cloud Service Account	8
References	12

Introduction

The following Report aims at clarifying the use of Virtual Machines for businesses who want to take advantage of their capabilities and upgrade their Data Infrastructures and daily procedures. The suggested provider is Google Cloud.

What is a Virtual Machine Instance (VM Instance)

A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual guest machines run on a physical host machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a MacOS virtual machine could run on a physical PC.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

An instance is a virtual machine (VM) hosted on Google's infrastructure. You can create an instance or create a group of managed instances by using the Google Cloud console, the Google Cloud CLI, or the Compute Engine API.


Compute Engine instances can run the public images for Linux and Windows Server that Google provides as well as private custom images that you can create or import from your existing systems. You can choose the machine properties of your instances, such as the number of virtual CPUs and the amount of memory, by using a set of predefined machine types or by creating your own custom machine types. Each instance belongs to a Google Cloud console project, and a project can have one or more instances. When you create an instance in a project, you specify the zone, operating system, and machine type of that instance. When you delete an instance, it is removed from the project. To create and manage instances, you can use a variety of tools, including the Google Cloud console, the gcloud command-line tool, and the REST API. To configure applications on your instances, connect to the instance using Secure Shell (SSH) for Linux instances or Remote Desktop Protocol (RDP) for Windows Server instances.

VM Instance Installation using Google Cloud


- Local OS: Windows 11
- VM Instance OS: Ubuntu

We visit the [Google Cloud webpage](#) and connect to our Google account. After that, we click on the Console. From there, we choose to create a new Project and we give it a name of our preference. Google offers 24 free projects to run per account as well as 300\$ of free credits for 3 months for every new account.


New Project

 You have 21 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
E-Commerce Shipping 

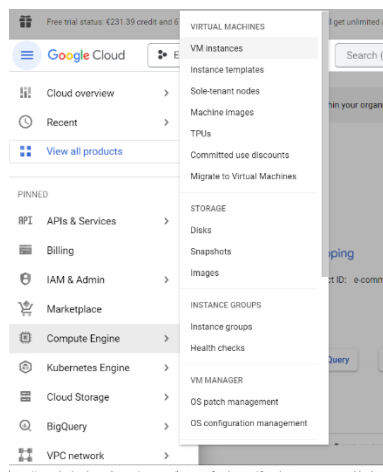
Project ID: e-commerce-shipping-379418. It cannot be changed later. [EDIT](#)

Location *
 No organization [BROWSE](#)

Parent organization or folder

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

Moving forward, we go Compute Engine -> VM Instance from the menu on the left.



We choose to create a new instance and we give it the name of our choice. We could also choose the region we want the server we connect to be located for better connectivity, the machine type to use as of its CPU and RAM requirements as well as the disk storage and the operating system. Here, we choose the Belgium region, on a 4 vCPU and 8gb RAM machine of 10gb disk storage, running Ubuntu.

Name *

ecommerce



! Name is already in use

Labels ?

+ ADD LABELS

Region *

europa-west1 (Belgium)



Region is permanent

Zone *

europa-west1-b



Zone is permanent

Machine type

e2-standard-4 (4 vCPU, 16 GB memory)



vCPU

4

Memory

16 GB

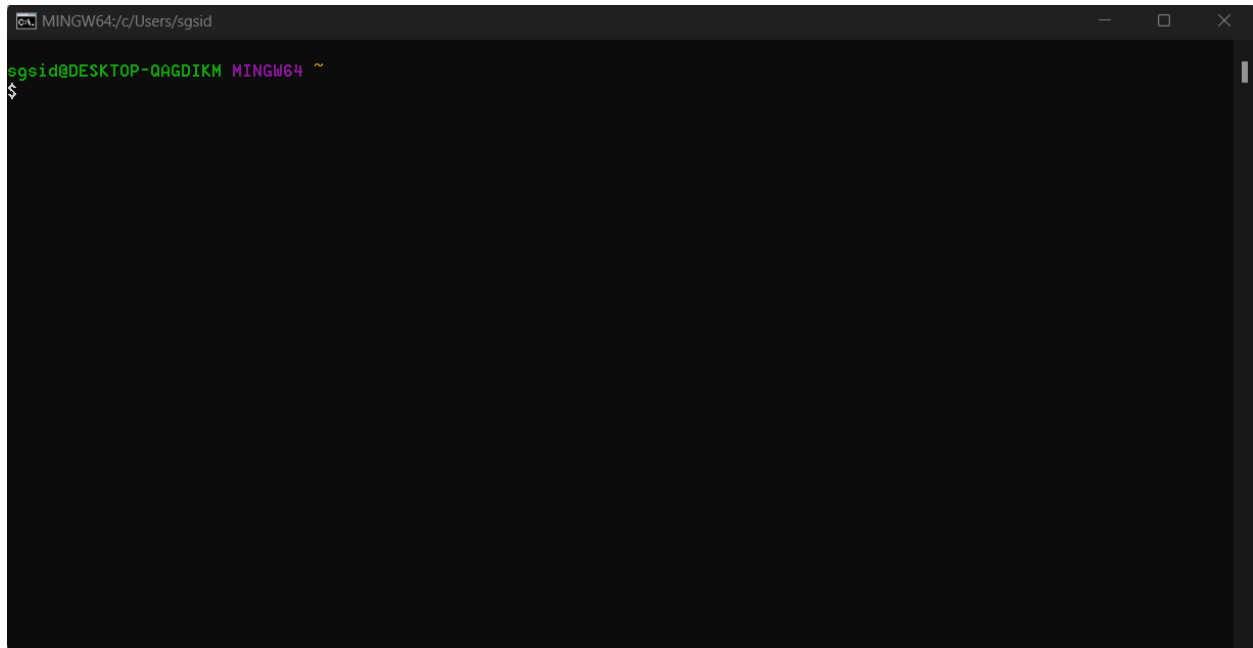
✓ CPU PLATFORM AND GPU

Boot disk ?

Name	ecommerce
Type	New balanced persistent disk
Size	10 GB
License type ?	Free
Image	Ubuntu 18.04 LTS

CHANGE

We continue on our terminal. Here, we use the GitBash terminal for windows offered by Git but you could use any other type of terminal in the operating system of your choice. Of course, some steps, paths and commands might vary depending on the operating system you are working with.




Firstly, we create a .ssh directory with the mkdir command. Here, we will store the ssh key. SSH keys are an authentication method used to gain access to an encrypted connection between systems and then ultimately use that connection to manage the remote system.



In order to generate the key we use the following command:

```
ssh-keygen -t rsa -f C:\Users\WINDOWS_USER\.ssh\KEY_FILENAME -C USERNAME -b 2048
```

where WINDOWS_USER is your username on the windows machine, KEY_FILENAME is the name for your SSH key file and USERNAME is your username in the VM.



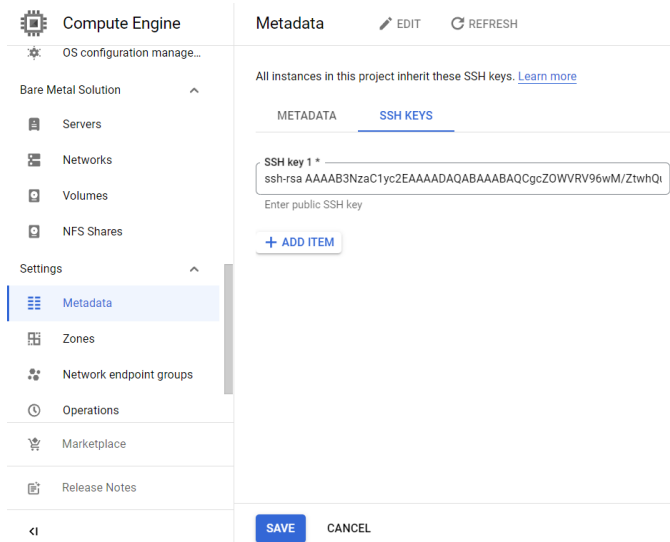
A private key file (gcp_1) and a public key file (gcp_1.pub) are generated inside the .ssh directory. Never share your private key file with anyone as it will provide someone with access to your VM instance. The one to be shared is the public key file gcp_1.pub where a hashed key is presented.



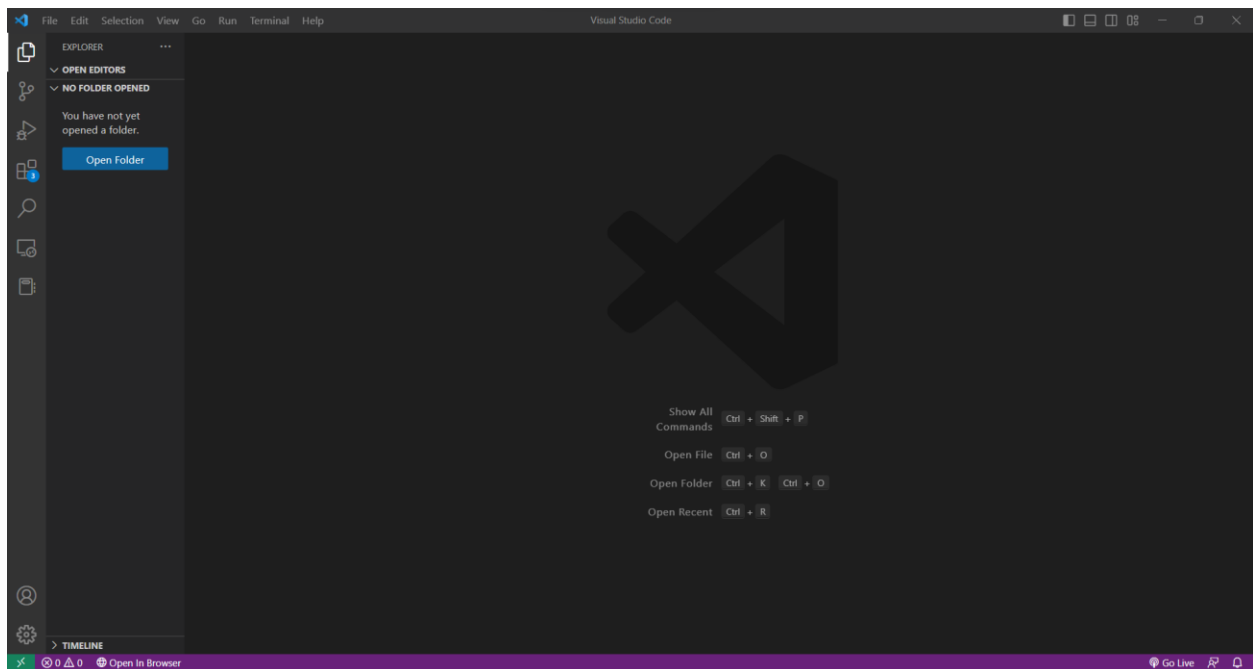
We open the public key file using the command cat and copy the public SSH key.

```
sgsid@DESKTOP-QAGDIKM MINGW64 ~/.ssh
$ cat gcp_1.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDoP16Hc5/FdrFiMhKD8t07KEcYnPimnAq8KzjujfibJ7Y7Q
j1PyJuI/U6CEhuq5uPRPLesH1c07iYeyz47Ya4FrrkSnuElxNw3qCQ+rOKYrbXKv6UCm8ZTieYPU+Z8119dCm
0uPUUp6GyTBWkaT5QqPcsMrn+QIyY25JiODGUNoMDzQUGco53r1WYu83dnk3uynai3m7k61R01JUdxqc4Ukm1
vFth0YokHuDYDt/UFN8/ stamatis
```

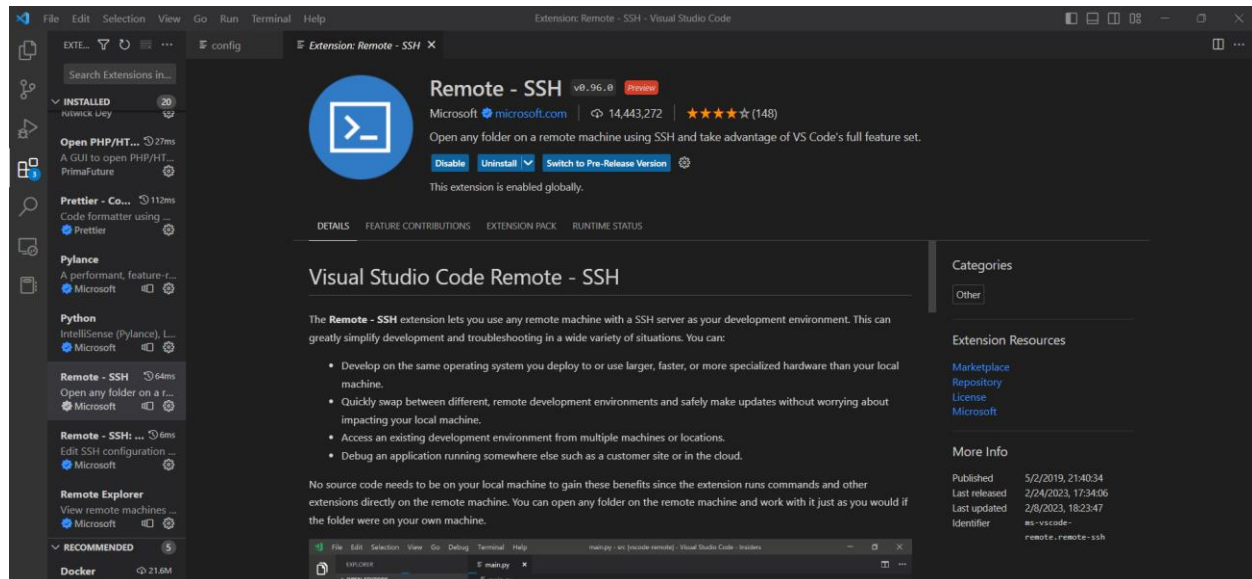
Next, we revisit our VM instance and this time we choose Metadata from the menu on the left. We choose the SSH KEYS option and add there our public SSH key we copied previously.



Now, we are ready to connect our local machine to the VM instance using the SSH network. To do so, we will use the code editor Visual Studio Code (VS).



We visit the extensions on the left and search for the Remote – SSH extension to install. The extension will help as configure the connection between our local machine and the virtual machine.

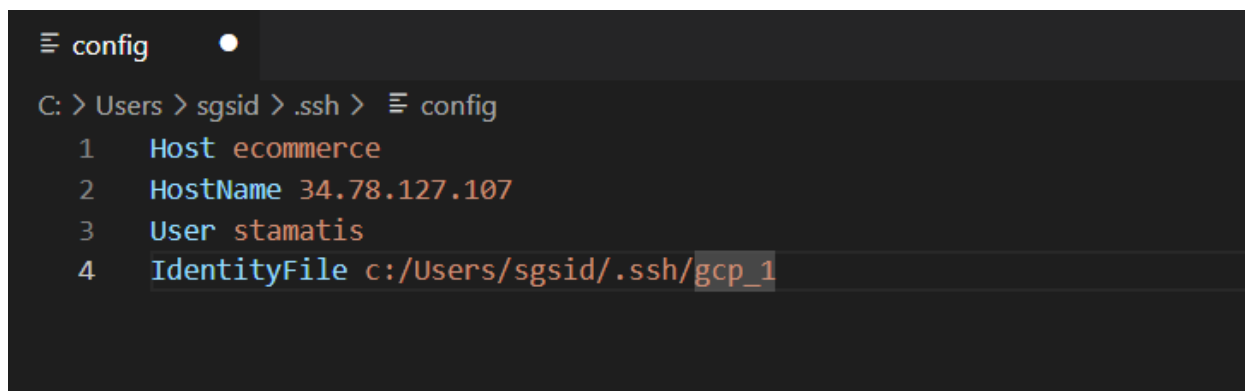


In order to configure we need a config directory that includes all the information needed. We create one via our terminal and open it with the command code.

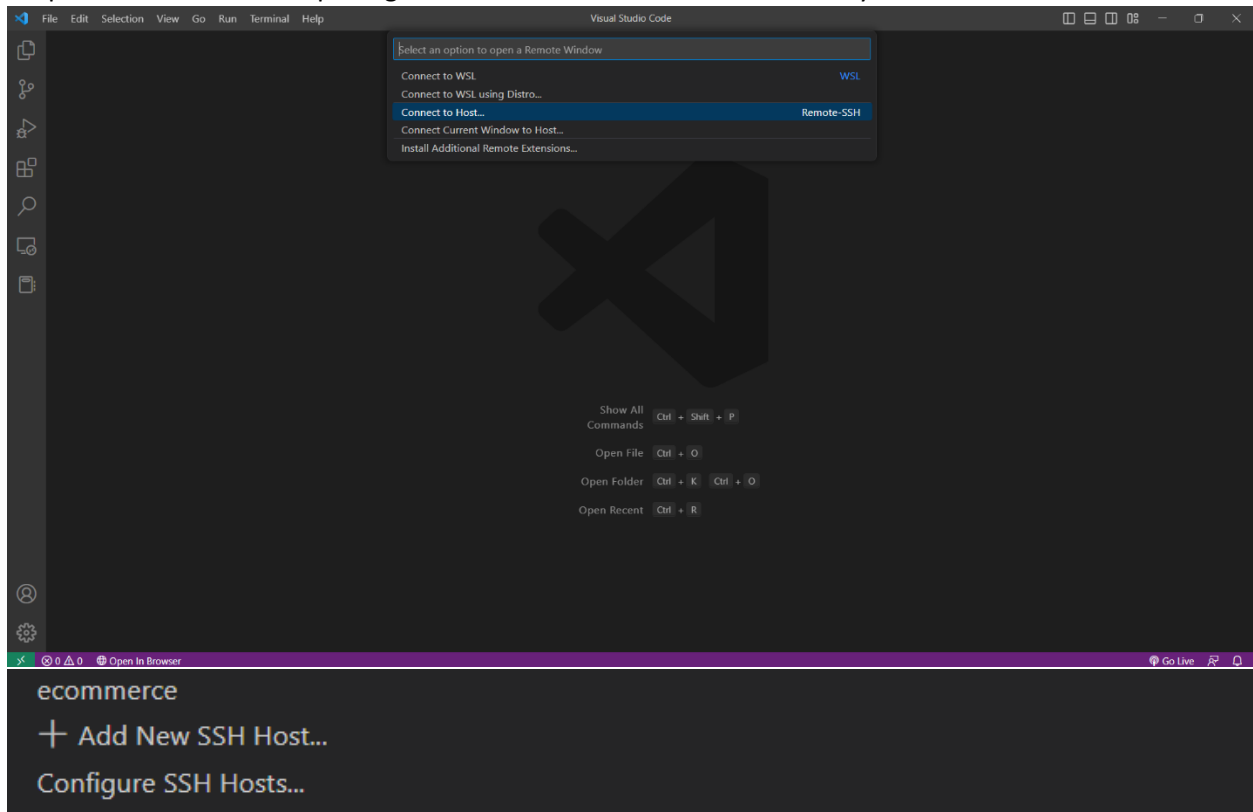
```
sgsid@DESKTOP-QAGDIKM MINGW64 ~/.ssh
$ mkdir config

sgsid@DESKTOP-QAGDIKM MINGW64 ~/.ssh
$ code config
```

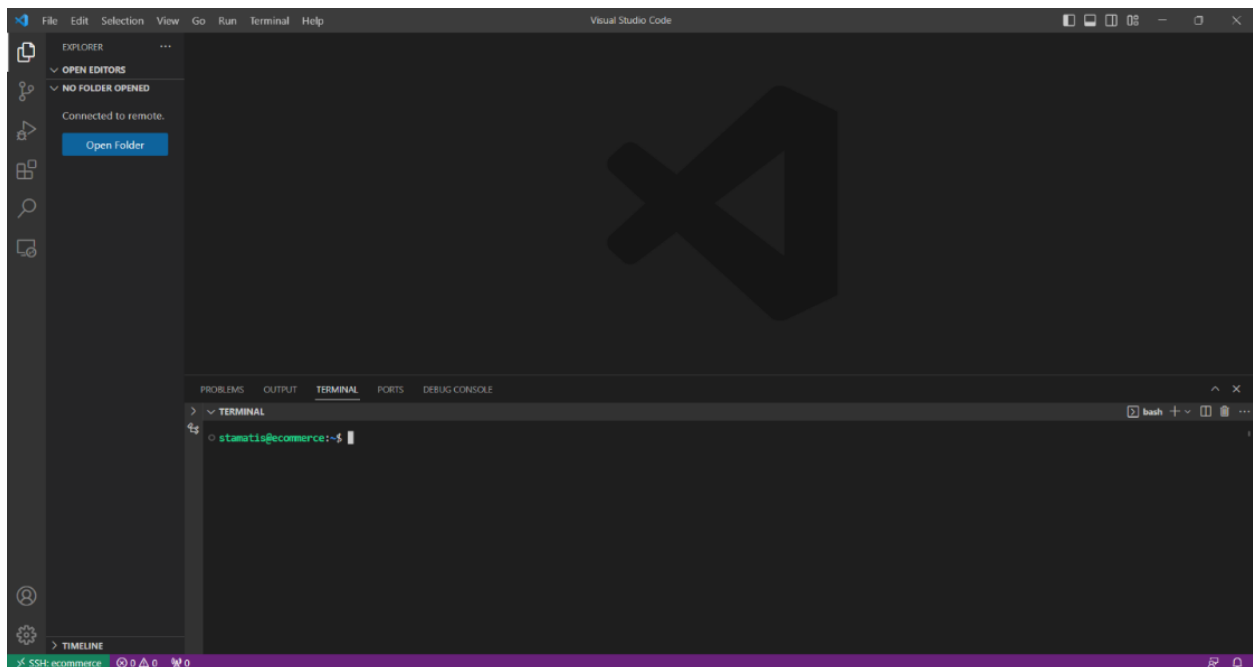
The directory opens via our default code editor which must be set as Visual Studio Code. Set the info with the following structure where Host is the name of the VM, HostName the external IP address of the VM, User the username you use as user for the VM and IdentityFile the local path to your private SSH key. Save the file.



Continue by choosing the green icon on the bottom left of VS and choose “Connect to host” from the drop-down menu opening. Choose the name of your Host and connect.



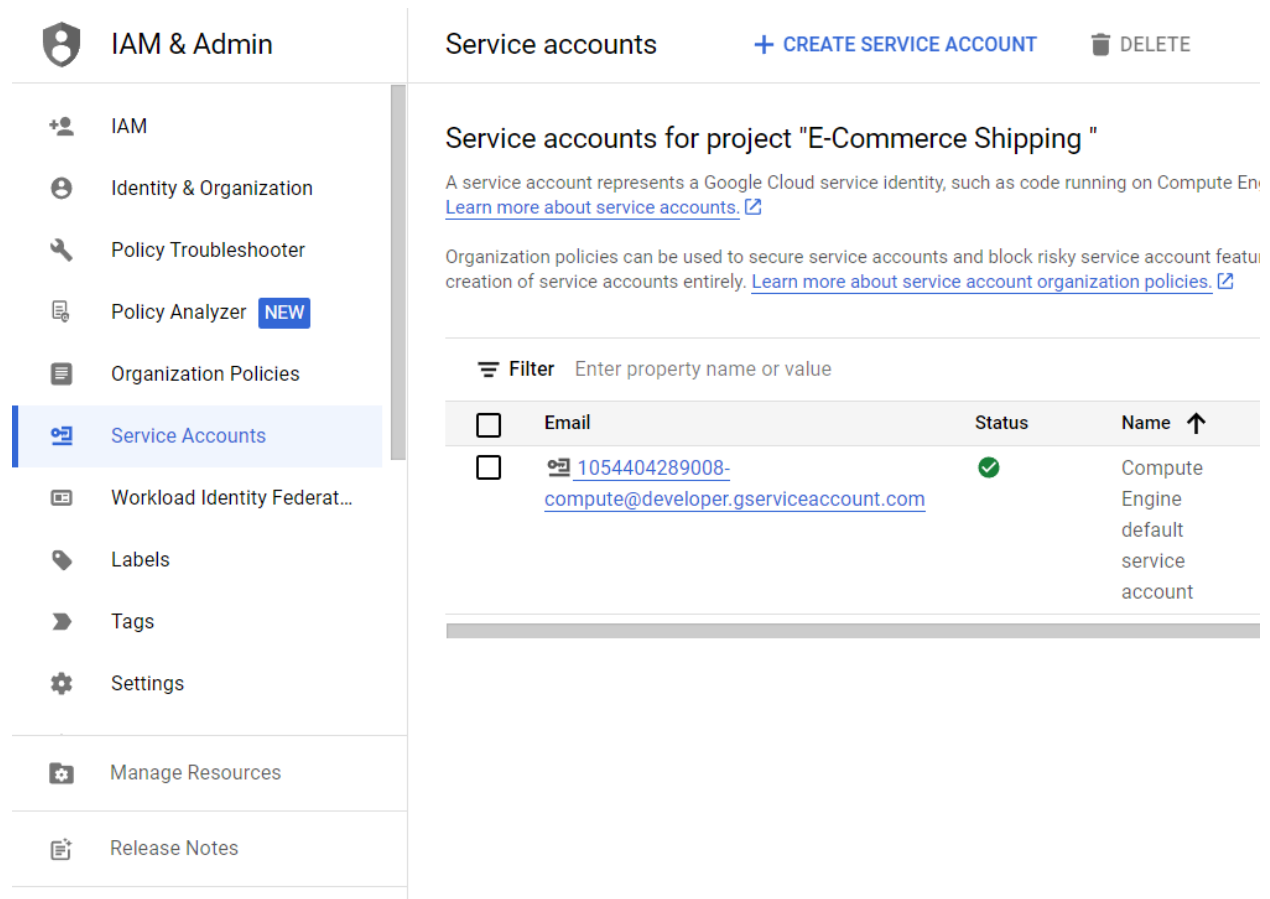
That’s it! You are connected to a Virtual Machine Instance of your preference provided by Google.



Creation of Google Cloud Service Account

The Service Account is a special type of Google account intended to represent a non-human user that needs to authenticate and be authorized to access data in Google APIs. Typically, service accounts are used in scenarios such as: Running workloads on virtual machines (VMs).

At first, we visit Google Cloud and choose the IAM & Admin -> Service Accounts -> CREATE SERVICE ACCOUNT from the menu on the left.



The screenshot shows the Google Cloud IAM & Admin console. On the left is a navigation menu with the following items: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (marked with a 'NEW' badge), Organization Policies, Service Accounts (highlighted with a blue bar), Workload Identity Federat..., Labels, Tags, and Settings. Below these are 'Manage Resources' and 'Release Notes'. The main content area is titled 'Service accounts' and includes a '+ CREATE SERVICE ACCOUNT' button and a 'DELETE' button. Below the title is a section for 'Service accounts for project "E-Commerce Shipping"' with explanatory text and links. A table below shows a list of service accounts with columns for checkboxes, Email, Status, and Name. One service account is listed with a green status icon.

	Email	Status	Name ↑
<input type="checkbox"/>	1054404289008-compute@developer.gserviceaccount.com	✓	Compute Engine default service account

Set a name of your choice, the Role to Viewer and create your account. The new Service Account should be displayed in Service Accounts page.

The screenshot shows the Google Cloud IAM & Admin console. The left sidebar lists navigation options: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federat..., Labels, Tags, Settings, and Manage Resources. The main content area is titled 'Service accounts' and shows a list of service accounts for project 'E-Commerce Shipping'. The list includes a table with columns: Email, Status, Name, Description, Key ID, Key creation date, and Actions. Two service accounts are listed: 'compute@developer.gserviceaccount.com' and 'ecommerce-user@e-commerce-379416.iam.gserviceaccount.com'. Both have a status of 'ON' and 'No keys'.

Email	Status	Name	Description	Key ID	Key creation date	Actions
compute@developer.gserviceaccount.com	ON	Compute Engine default service account		No keys		⋮
ecommerce-user@e-commerce-379416.iam.gserviceaccount.com	ON	ecommerce-user		No keys		⋮

We click the 3 bullets on the right of the service account we just created and choose the option “Manage Keys”. From there, we choose ADD KEY -> Create new key and we set the key type to JSON. This way we download locally the private key needed to use this service account.

Create private key for "ecommerce-user"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

As we are using a VM instance, we need to move the JSON file from our local environment to the VM to use it there. To do so, we visit again the Google Cloud and the VM instances option. We click on the SSH button of our VM instance and a SSH-in-browser window pops up. In case that more than one user exist in our Linux instance, it is possible that the Linux username presented differs from the one we want to use. We can change it by clicking on the settings button at the top right corner and selecting “Change Linux Username”. There we type our preferred username so that the uploaded files are uploaded to the

correct user. We click on UPLOAD FILE and upload the JSON file which is located in our default folder for downloading files.

The screenshot shows the SSH-in-browser interface. At the top, there's a header with 'SSH-in-browser' and buttons for 'UPLOAD FILE', 'DOWNLOAD FILE', and icons for notifications, keyboard shortcuts, and settings. The main terminal area displays system information as of Sat Mar 4 14:55:57 UTC 2023. It lists system load (0.0), usage of / (31.0% of 9.51GB), memory usage (3%), swap usage (0%), processes (150), users logged in (1), and IP address for ens4 (10.132.0.2). There are also announcements about Kubernetes security and Ubuntu Pro subscription. A sidebar on the right contains settings like Theme, Font, Font size, Copy preference, Keyboard preference, Show Scrollbar, New Connection, Change Linux Username, and Instance Details. At the bottom, a confirmation box shows 'Transferred 1 item' with a dropdown arrow and a close button, and a list item 'e-commerce-shipp...' with a green checkmark.

```
System information as of Sat Mar  4 14:55:57 UTC 2023

System load:  0.0          Processes:           150
Usage of /:   31.0% of 9.51GB Users logged in:       1
Memory usage: 3%          IP address for ens4: 10.132.0.2
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/gcp/pro

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

stamatis@ecommerce:~$
```

Transferred 1 item

e-commerce-shipp...

Back in our VM's terminal, we create a new directory called keys and move the private key there for organization reasons.

```
stamatis@ecommerce:~$ ls
e-commerce-shipping-379416-1deae3e0742f.json  keys  snap
stamatis@ecommerce:~$ mv e-commerce-shipping-379416-1deae3e0742f.json keys
stamatis@ecommerce:~$ ls
keys  snap
```

We will now use the Google Cloud Client to configure our connection to the service account. Our VM instance already includes Google Cloud Client and so there is no need to download it. By using the following command, we set the environment variable to point to our downloaded GCP auth-key and then we login to the service account.

```
stamatis@ecommerce:~$ export GOOGLE_APPLICATION_CREDENTIALS="keys/e-commerce-shipping-379416-1deae3e0742f.json"
stamatis@ecommerce:~$ gcloud auth application-default login
```

We then follow the instructions displayed and in the end, we should be connected to our service account.

```
Credentials saved to file: [/home/stamatis/.config/gcloud/application_default_credentials.json]

These credentials will be used by any library that requests Application Default Credentials (ADC).

Quota project "e-commerce-shipping-379416" was added to ADC which can be used by Google client libraries for billing and quota. Note that some services may still bill the project owning the resource.
```

Back to Google Cloud, we are able to change the roles and authorization our user has in order to allow them have access to different services of the cloud. To do so, we visit our service account by going to IAM & Admin -> IAM and we choose the pencil button next to our service account. There, for example, we

choose to add 3 new roles, one for Storage Admin, one for Storage Object Admin and one for BigQuery Admin.

Edit access to "E-Commerce Shipping "

Principal













ecommerce-user@e-commerce-shipping-379416.iam.gserviceaccount.com

Project

E-Commerce Shipping

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

<div>Role </div> <div>BigQuery Admin</div> <div>Administer all BigQuery resources and data</div>	<div>IAM condition (optional) </div> <div>+ ADD IAM CONDITION</div>	
<div>Role </div> <div>Storage Admin</div> <div>Full control of GCS resources.</div>	<div>IAM condition (optional) </div> <div>+ ADD IAM CONDITION</div>	
<div>Role </div> <div>Storage Object Admin</div> <div>Full control of GCS objects.</div>	<div>IAM condition (optional) </div> <div>+ ADD IAM CONDITION</div>	
<div>Role </div> <div>Viewer</div> <div>View most Google Cloud resources. See the list of included permissions.</div>	<div>IAM condition (optional) </div> <div>+ ADD IAM CONDITION</div>	

+ ADD ANOTHER ROLE

SAVE

TEST CHANGES



CANCEL

As a final step, we need to enable two APIs found here:

<https://console.cloud.google.com/apis/library/iam.googleapis.com?project=ivory-lotus-374512>

<https://console.cloud.google.com/apis/library/iamcredentials.googleapis.com?project=ivory-lotus-374512>

They manage identity and access control for Google Cloud Platform resources, including the creation of service accounts, which you can use to authenticate to Google and make API calls.

References

<https://www.vmware.com/topics/glossary/content/virtual-machine.html>

<https://cloud.google.com/compute/docs/instances>

<https://cloud.google.com/iam/docs/service-account-overview>