

Assignment: Exploring Google Cloud Services

Kaimakova Aigerim

17.10.2024

[GitHub](#)

Table of Contents

1. Introduction	3
2. Virtual Machines in Google Cloud	4
2.1 VM Creation	4
2.2 Connection	4
2.3 Findings	5
3. Storage Solutions in Google Cloud	5
3.1 Bucket Creation	5
3.2 Object Lifecycle Management	6
3.3 Findings	7
4. Networking in Google Cloud	7
4.1 VPC Setup	7
4.2 Connectivity	7
4.2 Findings	10
5. Conclusion	10
6. References	10

Introduction

This assignment focuses on exploring key services provided by **Google Cloud Platform (GCP)**, specifically in the areas of virtual machines, cloud storage, and networking. The primary objective is to gain practical experience with setting up and managing cloud resources, including the creation of a Virtual Machine (VM) instance, configuring a web server, utilizing cloud storage, and establishing networking through Virtual Private Cloud (VPC).

By completing this assignment, students will develop skills in:

- Deploying and configuring cloud-based virtual machines.
- Installing and running web servers on cloud infrastructure.
- Managing cloud storage buckets and setting lifecycle rules.
- Setting up and securing network configurations for virtual machines using VPC and firewall rules.

The assignment provides hands-on experience with cloud technologies that are essential for managing scalable and efficient cloud-based applications and infrastructure.

1. Virtual Machines in Google Cloud

VM Creation:

To create a Virtual Machine (VM) in Google Cloud, I went to the **Compute Engine** section and clicked on **VM instances**. Then, I clicked **Create Instance** and chose the machine type. I selected a small, basic machine type to keep it simple and picked **Ubuntu** as the operating system. In the firewall section, I checked the box to allow **SSH** traffic so that I could connect to the VM. After everything was set, I clicked **Create** to launch the VM.

instance-20241015-190619	asia-east2-b	10.170.0.2 (nic0)	34.92.5.80 (nic0)	SSH ▾
<div>← instance-2024... EDIT RESET + CREATE MACHINE</div>				
instance-20241015-190619				
<div>DETAILS OBSERVABILITY OS INFO SCREENSHOT</div>				
Basic information				
Name	instance-20241015-190619			
Instance Id	1161329262685552610			
Description	None			
Type	Instance			
Status	✔ Running			
Creation time	Oct 16, 2024, 12:22:59 AM UTC+05:00			
Location ?	asia-east2-b			
Instance template	None			
In use by	None			
Reservations	Automatically choose			
Labels	None			
Tags ?	—			
Deletion protection	Disabled			
Confidential VM service ?	Disabled			
Preserved state size	0 GB			

Fig 1. Basic information about created VM

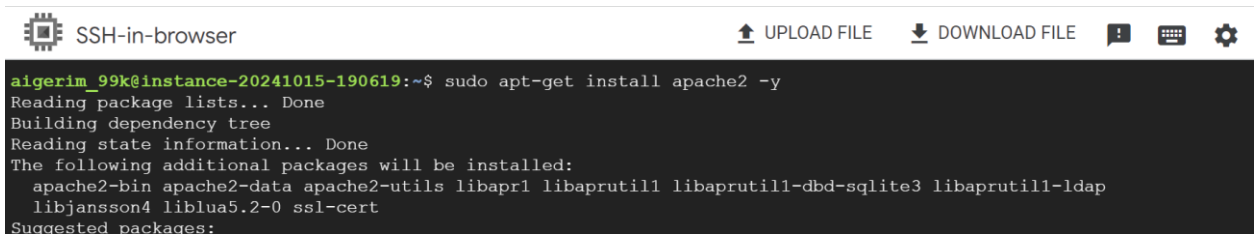
Connection:

Once the VM was running, I connected to it by clicking the **SSH** button next to the instance name. This opened a terminal in my browser. From there, I installed a web server. I first updated the package list using the command:

```
https://ssh.cloud.google.com/v2/ssh/projects/genuine-plate-437009-k1/zones/asia-east2-b/instances/instance-20241015-190619?a...
SSH-in-browser
aigerim_99k@instance-20241015-190619:~$ sudo apt-get update
Hit:1 http://asia-east2.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://asia-east2.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 http://asia-east2.gce.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
```

Fig 2. Command for updating the package list

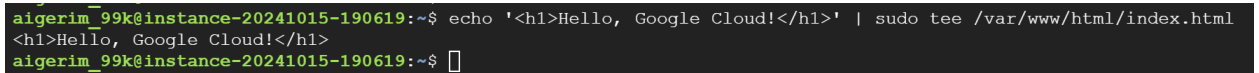
Then, I installed Apache by running:



```
aigerim_99k@instance-20241015-190619:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libjansson4 liblua5.2-0 ssl-cert
Suggested packages:
```

Fig 3. Command for Apache installation

After that, I created a simple HTML file to test the web server. I used this command to create a test page:



```
aigerim_99k@instance-20241015-190619:~$ echo '<h1>Hello, Google Cloud!</h1>' | sudo tee /var/www/html/index.html
<h1>Hello, Google Cloud!</h1>
aigerim_99k@instance-20241015-190619:~$
```

Fig 4. Command for creating a HTML page

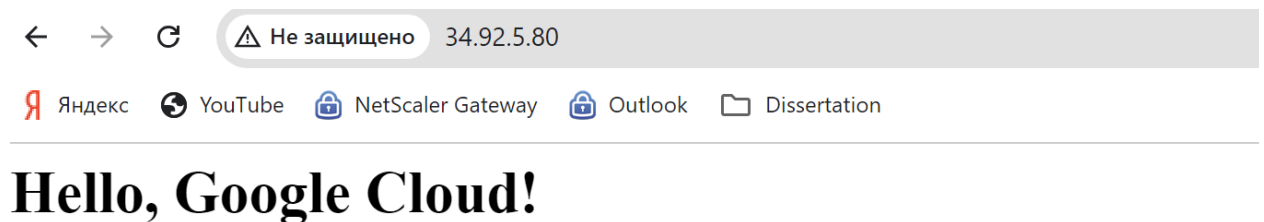


Fig 5. Test HTML page

Findings:

Everything worked fine, and I could access the test page by typing my VM's external IP into a browser. The page showed "Hello, Google Cloud!" as expected. One small challenge I faced was the error with the exclamation mark when creating the HTML file, but I quickly fixed it by using single quotes around the text. Overall, the process was straightforward, and I learned how to set up and connect to a VM, as well as install a web server.

2. Storage Solutions in Google Cloud

Bucket Creation:

To create a storage bucket in Google Cloud, I first went to the **Storage** section by clicking on **Buckets** under the **Storage** tab in the Google Cloud Console. I clicked **Create Bucket** and chose a unique name **bucket-aassignment2** for my bucket. I then selected the **location** (I picked a multiple regions in Asia) and the **storage class** as **Standard** because it's suitable for frequently accessed data. For access controls, I decided to make the bucket **private**, as I didn't want the files to be publicly accessible.

← Create a bucket

- Get Started**

Pick a globally unique, permanent name. [Naming guidelines](#)

bucket-aassignment2

Tip: Don't include any sensitive information

Optimize storage for data-intensive workloads

Labels (optional)

CONTINUE
- Choose where to store your data**

This choice defines the geographic placement of your data and affects cost, performance, and availability. Cannot be changed later. [Learn more](#)

Location type

☒ Multi-region
Highest availability across largest area

asia (multiple regions in Asia)

☐ Add cross-bucket replication via Storage Transfer Service

Go to next step

Fig 6. Creating a bucket

Upload a Sample File:

After creating the bucket, I clicked on it and used the **Upload Files** button to add a sample file (I chose an image). Once the file was uploaded, it appeared in the bucket and was ready to be accessed based on the access control settings I had chosen earlier.

bucket-aassignment2

Location	Storage class	Public access	Protection
asia (multiple regions in Asia)	Standard	Not public	Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Buckets > bucket-aassignment2

CREATE FOLDER UPLOAD TRANSFER DATA OTHER SERVICES

Filter by name prefix only

Upload files
Upload folder

Name	Size	Type	Created	Storage class	Last modified	Public access
2. Хаски.jpg	327.5 KB	image/jpeg	Oct 16, 2024, 1:35:48 AM	Standard	Oct 16, 2024, 1:35:48 AM	Not public

1 file successfully uploaded

Fig 7. Uploading image

Object Lifecycle Management:

Next, I set up **Lifecycle Management** for the bucket. This feature automatically manages the lifecycle of objects in the bucket, such as deleting them after a certain period. To set this up, I went to the **Bucket Settings** and under **Lifecycle Management**, I added a rule to **delete objects after 30 days**. This means that any file I upload to the bucket will be automatically deleted after 30 days.



Fig 10. VPC creation

vpc-ass2

< OVERVIEW **SUBNETS** STATIC INTERNAL IP ADDRESSES FIREWALLS FIREWALL >

Subnets [+ ADD SUBNET](#) [MANAGE FLOW LOGS](#)

Filter Enter property name or value

<input type="checkbox"/>	Name ↑	Region	Stack Type	Primary IPv4 range	Secondary IPv4 range	
<input type="checkbox"/>	subnet-asia-east2	asia-east2	IPv4	10.1.0.0/24		 

Create a firewall rule

Name *

allow-http-https

?

Lowercase letters, numbers, hyphens allowed

Description

Logs

Turning on firewall logs can generate a large number of logs which can increase costs in Logging. [Learn more](#)

☐ On

☒ Off

Network *

vpc-ass2

▼

?

Priority *

1000

COMPARE

?

Priority can be 0 - 65535

Direction of traffic

?

☒ Ingress

☐ Egress

Action on match

?

☒ Allow

☐ Deny

8

← Firewall rule details	← Firewall rule details
allow-http-https	allow-ssh
Logs ? Off view in Logs Explorer	Logs ? Off view in Logs Explorer
Network vpc-ass2	Network vpc-ass2
Priority 1000	Priority 1000
Direction Ingress	Direction Ingress
Action on match Allow	Action on match Allow
Source filters	Source filters
IP ranges 0.0.0.0/0	IP ranges 0.0.0.0/0
Protocols and ports tcp:80, 443	Protocols and ports tcp:22
Enforcement Enabled	Enforcement Enabled

Fig 13-14. Firewall Rules that allows HTTP,HTTPS,SSH traffic

Connectivity: To make sure my VM could connect to the internet, I assigned it an external IP address. Then, I checked the firewall rules to ensure they allowed incoming traffic on necessary ports, such as port 22 for SSH, and ports 80 and 443 for web traffic. After confirming the settings, I tested the connection by trying to ping an external server and accessing my VM via SSH. This confirmed that my VM was able to communicate with the internet properly.

```
Last login: Tue Oct 15 19:41:49 2024 from 35.235.243.97
aigerim_99k@instance-20241015-190619:~$ ping google.com
PING google.com (142.250.76.238) 56(84) bytes of data:
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=1 ttl=122 time=3.32 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=2 ttl=122 time=0.899 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=3 ttl=122 time=0.932 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=4 ttl=122 time=0.874 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=5 ttl=122 time=0.869 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=6 ttl=122 time=0.829 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=7 ttl=122 time=0.871 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=8 ttl=122 time=0.825 ms
64 bytes from nchkga-ad-in-f14.1e100.net (142.250.76.238): icmp_seq=9 ttl=122 time=0.865 ms
```

Fig 15. Connectivity Test with ping command

Network interfaces ?

Network interface is permanent

^ Edit network interface

Network *

vpc-ass2

▼ ?

Subnetwork *

subnet-asia-east2 IPv4 (10.1.0.0/24)

▼ ?

i

To use IPv6, you need an IPv6 subnet range. [LEARN MORE](#)

Findings: Setting up networking in the cloud is important because it lets me control how my resources are organized and secured. A VPC gives me an isolated environment, and firewall rules protect my VMs by allowing only the traffic I need. Cloud networking is essential for making sure everything works smoothly and safely, especially when connecting VMs to the internet.

Conclusion

In this assignment, I learned how to use different Google Cloud services like Virtual Machines, Cloud Storage, and Virtual Private Cloud (VPC). I learned how to create and manage a virtual machine, set up a storage bucket to save files, and create a private network to control how my resources connect to the internet. The process showed me how cloud services can help organize and secure data and make it easier to manage applications.

These Google Cloud services are useful for many things, like hosting websites, storing files safely, and managing networks. They allow businesses to work more efficiently and securely, without needing physical servers or hardware. I can see how these tools could be used for building apps, storing large amounts of data, or creating secure networks for businesses.

References

1. Google Cloud documentation: <https://cloud.google.com/docs>
2. Google Cloud tutorials for Compute Engine and Cloud Storage setup
3. Various online resources and guides on networking and cloud infrastructure