KAZAKH BRITISH TECHNICAL UNIVERSITY

REPORT

Assignment 2, Cloud Computing

Akhmetkan Azhar
24MD0470
17.10.2024

Table of Contents

Introduction

This assignment aims to give hands-on experience with Google Cloud services, with particular emphasis on virtual machines (VMs), storage options and networking setups. Through the process of establishing and customizing these services, we will get more profound comprehension of cloud infrastructure and its uses. We will record our results and consider potential and difficulties that come with using Google Cloud throughout the assignment. The goal of this assignment approach is increasing knowledge of cloud tools and show how they can be used for practical IT solutions.

Virtual Machines in Google Cloud

- **Create a Virtual Machine (VM) Instance**
  - Use the Google Cloud Console to create a VM instance.
  - Select an appropriate machine type, operating system, and region.
  - Configure the firewall to allow SSH traffic.
- **Connect to the VM**
  - Use the SSH option from the Cloud Console to connect to your VM.
  - Install a web server (e.g., Apache or Nginx) on the VM.
  - Create a simple HTML page to serve as a test.
- **Document the Process**
  - Take screenshots of each step: instance creation, connection, and web server setup.
  - Write a brief explanation of the choices made during setup.

Virtual Machine creation

To launch a virtual machine instance in Google Cloud Console, I first navigated to the Compute Engine section, enabled it, and added a billing account(Image 1.1). After that, I clicked on "Create Instance" and proceeded to select the desired operating system, location, and machine type to configure the virtual machine(Image 1.2). Once I reviewed and confirmed the settings, I clicked the "Create" button, successfully starting the virtual machine(Image 1.3). At last, my new virtual machine instance was ready(Image 1.4).

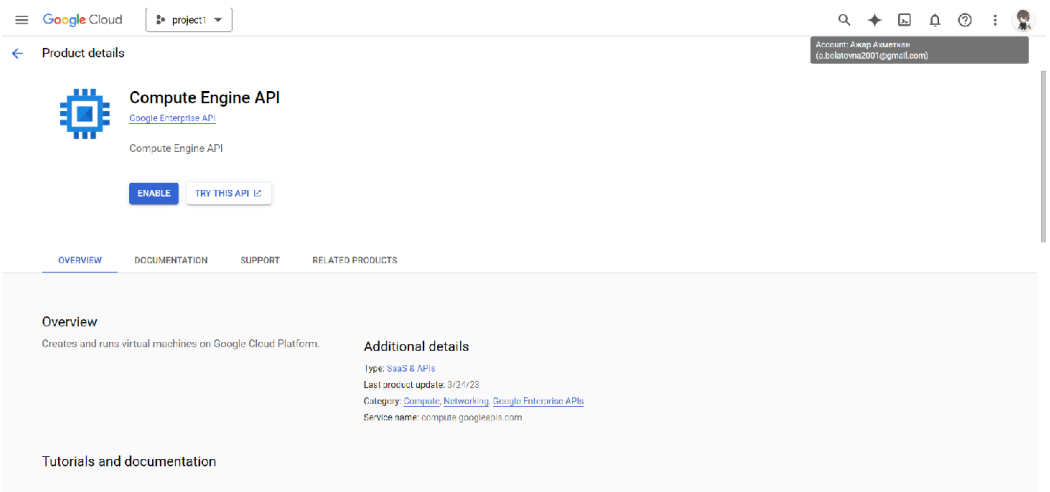


Image1.1

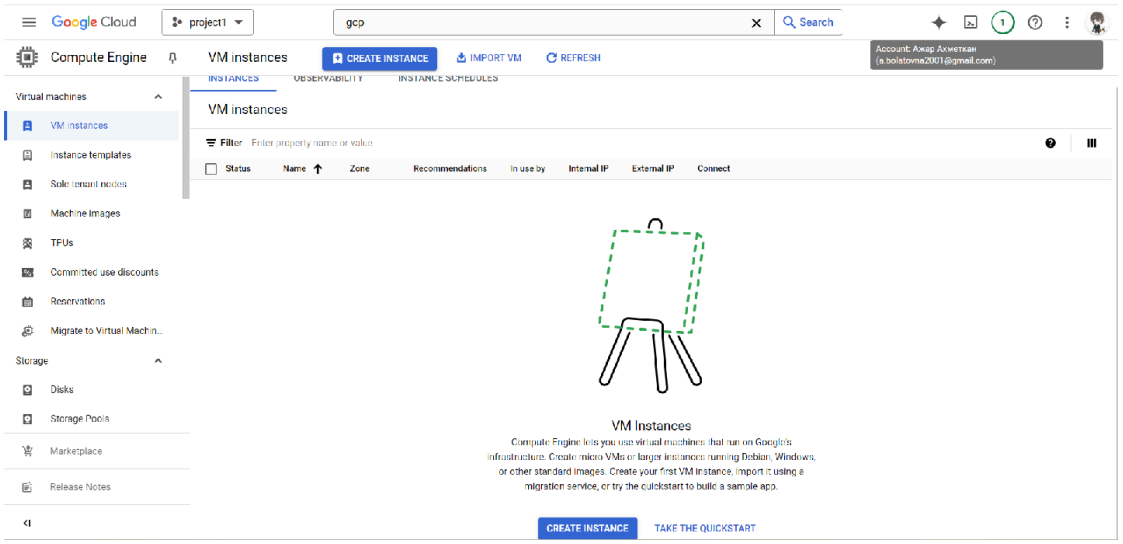


Image 1.2

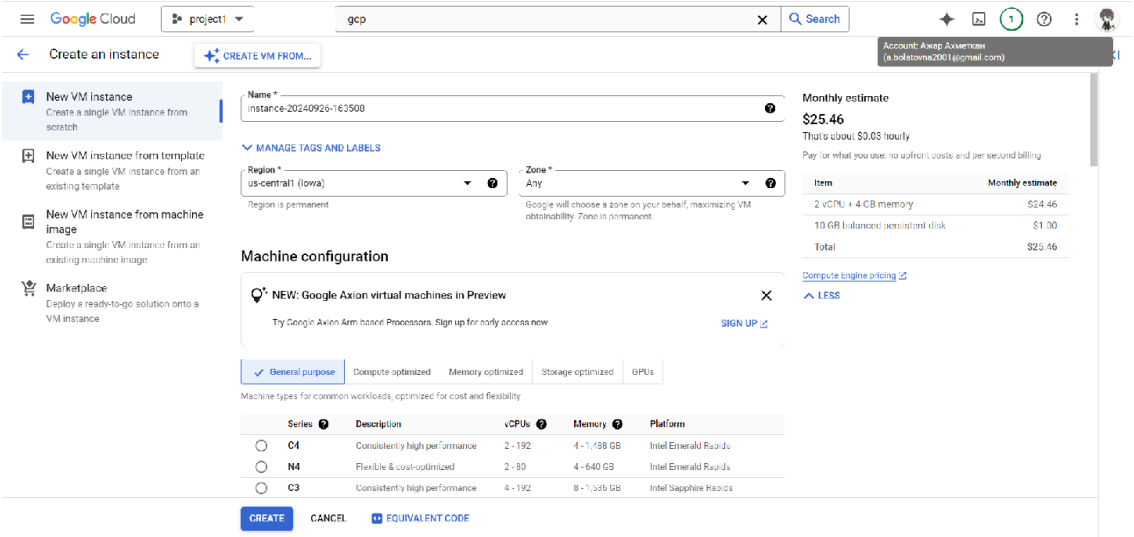


Image 1.3

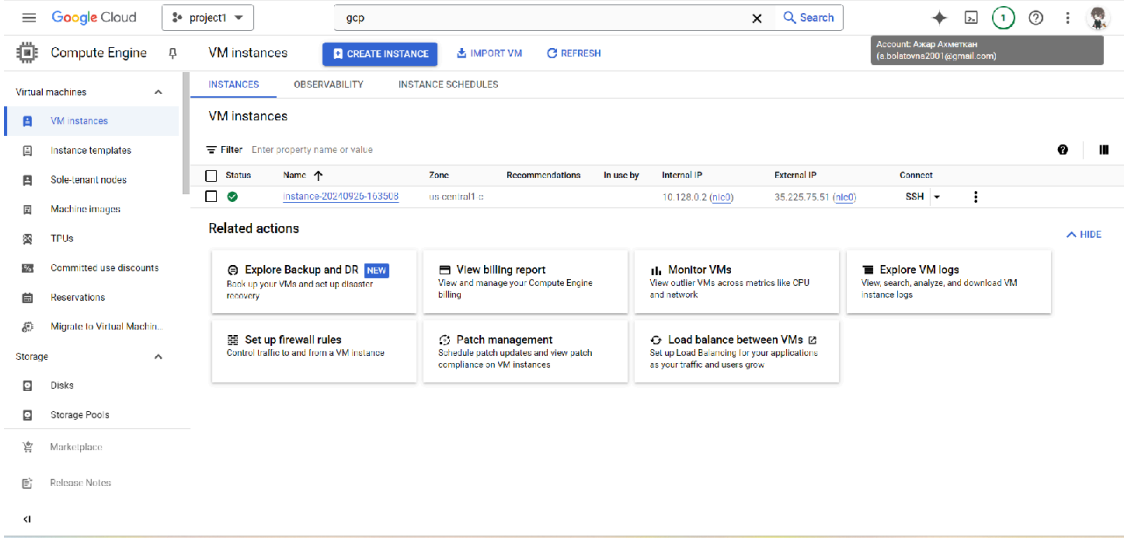


Image 1.4

Connection

I utilized the SSH option that was accessible straight from Google Cloud Console to establish connection with the VM. After establishing a connection with the VM using the SSH interface, I carried out the web server installation. In this instance, I used the following command to install Apache:

sudo apt-get install apache2

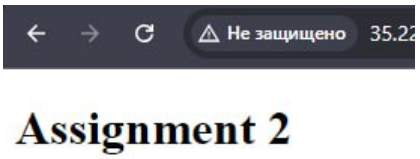
Following installation, I made a basic HTML page to use as a test. I used the following command to browse to the default web directory:

cd /var/www/html

Next, I composed the following text in an HTML file:

```
<!DOCTYPE html>
<html>
        <body>
                <h1>Assignment 2</h1>
         </body>
</html>
```

I made sure Apache was operating and saved the file. I then verified that the web server had been installed successfully and that the page was being served by using the VM's external IP address to access the test page in a web browser(Image 1.5).



Assignment 2

Image 1.5

Findings

I have positive outcomes and valuable lessons by working with Virtual Machines. I successfully created a VM with the desired configuration and installed a web server , which allowed me to serve content via a simple HTML page. This demonstrated the effectiveness of Google Cloud for hosting applications and improved my navigation skills in the Cloud Console.

However, I faced several challenges. Initially, I encountered confusion over machine type selection and struggled with configuring firewall rules to allow HTTP/HTTPS traffic. I also faced billing issues when I didn’t properly set up a billing account and missed my money, which temporarily halted my

progress. This experience underscored the importance of understanding billing requirements and monitoring costs in cloud environments.
It reinforced the capabilities of cloud infrastructure for deploying applications and provided key insights into configuration, security, resource management, and billing, preparing me for future cloud work.

Storage Solutions in Google Cloud

- **Create a Cloud Storage Bucket**
  - Use the Google Cloud Console to create a Cloud Storage bucket.
  - Set the bucket's access controls (public/private).
  - Upload a sample file (e.g., image or document) to the bucket.
- **Implement Object Lifecycle Management**
  - Set up a lifecycle rule for your bucket to automatically delete objects after a certain period (e.g., 30 days).
- **Document the Process**
  - Take screenshots of the bucket creation, file upload, and lifecycle rule setup.
  - Explain the use cases for Cloud Storage and the benefits of lifecycle management.

Bucket Creation

I visited the Cloud Storage area of the Google Cloud Console in order to build a Cloud Storage bucket (Image 2.1). I then select "Create" next to buckets. I set up my bucket, gave it a distinctive name, chose a location, a storage type, and an access control mechanism (Image 2.2). I made a create click and acquired a fresh bucket(Image 2.2).

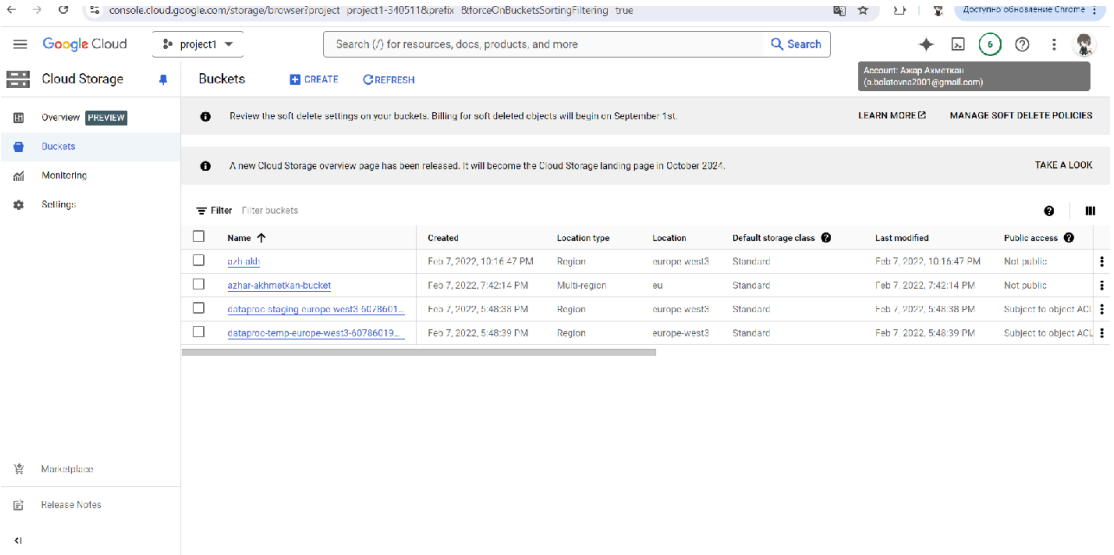


Image 2.1

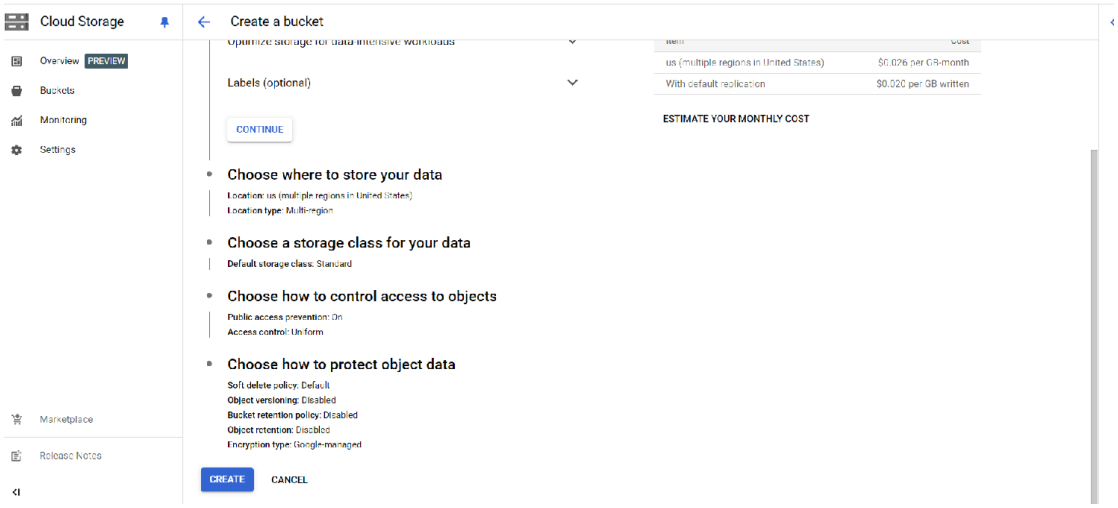


Image 2.2

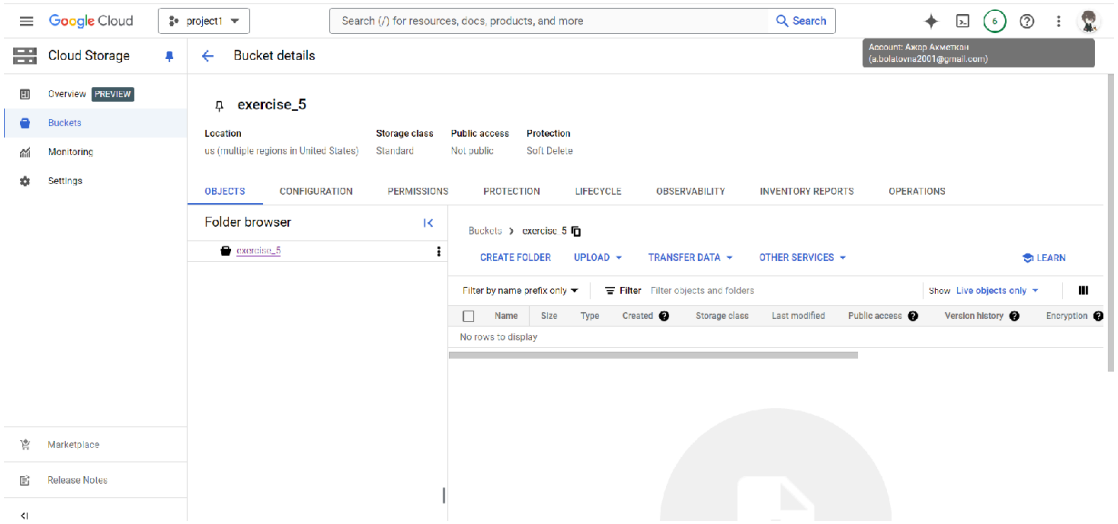


Image 2.3

Then I can set up the bucket's access restrictions. I could have made the information public, which is accessible to anybody with the link or private, which is accessible only to certain users or roles for this task. You also can change your accessibility after bucket creation.

I can add a sample file, like an image or document, using the "Upload Files" option after the bucket has been created. After the upload, the file showed up in the bucket, where I could adjust its permissions or share it based on the access control settings of the bucket.

Lifecycle Management

I can create a lifecycle rule to control the bucket's content over time once I upload the file.

I can click "Add a Rule" after navigating to the bucket's "Lifecycle" settings in the console.

By giving specifics to the action “Delete” and condition - 30 days after creation. I can set up the rule to automatically destroy objects after 30 days.

I can save the configuration after the rule is established. This optimizes storage management by guaranteeing that things saved in the bucket are automatically deleted after 30 days.

This process allowed me to create and manage a storage solution while automating file deletion, showcasing Google Cloud's powerful storage capabilities.

Findings

I got significant insights and practical outcomes by working with Cloud Storage in Google Cloud. I successfully created a Cloud Storage bucket, configured access controls, and uploaded a sample file, demonstrating the platform's user-friendly interface for managing data.

However, I faced some challenges during this process. Setting the appropriate access controls for the bucket was initially confusing, as I needed to ensure that the bucket was secure while allowing the necessary access for specific users. Additionally, I learned about Object Lifecycle Management when implementing rules to automatically delete files after a specified period. Although this feature is beneficial for managing storage costs, I encountered some difficulties in understanding the syntax and criteria for the rules.

Overall, this experience enhanced my understanding of data management in the cloud. It highlighted the importance of effective access control and automated data management features, which can help organizations reduce costs and streamline operations. I now feel more confident in utilizing Google Cloud Storage for future projects.

Networking in Google Cloud

- **Set Up a Virtual Private Cloud (VPC)**
  - Create a new VPC network with subnets.
  - Configure firewall rules to allow traffic between your VM and the internet.
- **Connect VM to VPC**
  - Ensure the VM created in the previous section is connected to the new VPC.
  - Use the VM to ping an external server to verify connectivity.
- **Document the Process**
  - Take screenshots of VPC creation, firewall configuration, and connectivity tests.
  - Discuss the importance of VPCs and firewall rules in cloud networking.

VPC Setup

I went to the Google Cloud Console's VPC Networks section to start a new Virtual Private Cloud (VPC). I chose "Create VPC Network" and gave the network a name. I configured subnets in the following step by giving each subnet's IP range and region. To enable traffic management and regional segmentation, I built several subnets.
I clicked the "Create" button to start the VPC after the subnets were defined.

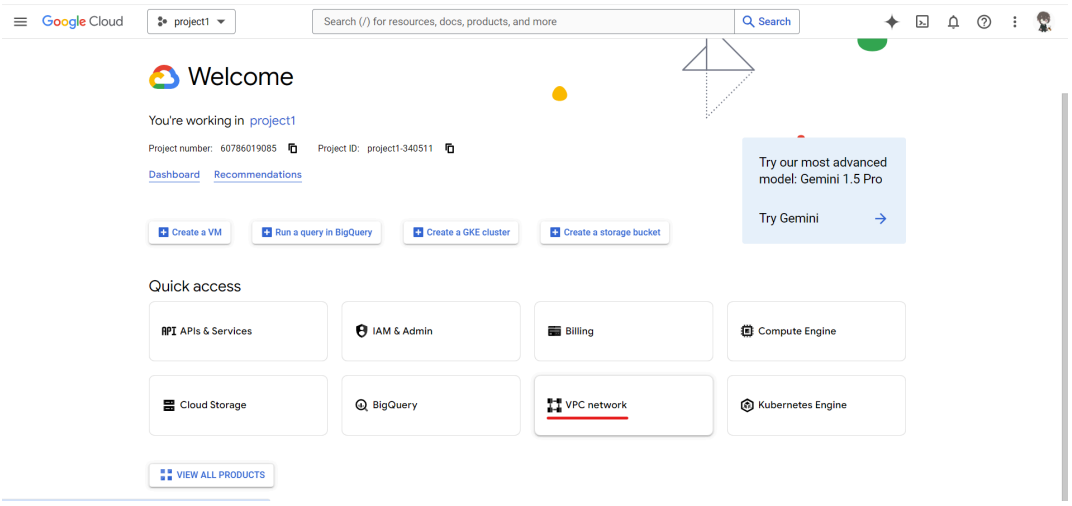


Image 3.1

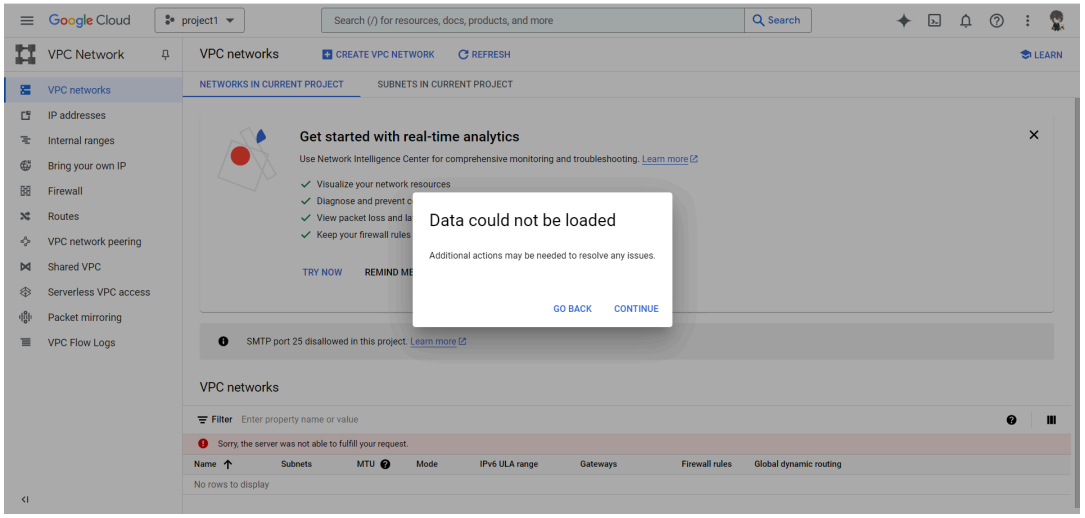


Image 3.2

I can set up firewall rules to control in and out traffic after configuring the VPC.

Connectivity

To make sure my virtual machine can deliver online content, I can make rule in the VPC network's Firewall Rules section that permits HTTP traffic on port 80 and HTTPS traffic on port 443.

Also, in order to make pinging and network connectivity tests easier, I can add rules that permit ICMP packets. I can make sure that traffic between the virtual machine and internet was permitted and that rules were implemented for the VPC network.

I will verify the VM's network interface settings in Compute Engine section once I have set up the VPC to make sure the previously built VM was linked to the new VPC. In order to enable the VM to use the custom subnets and firewall rules I previously set, it will be connected to the new VPC network.

I can access the virtual machine using the SSH terminal from the Google Cloud Console to confirm connectivity. To verify network connectivity, I may use the VM to do a basic ping command to an external server, like

ping 8.8.8.8

Google's public DNS server. The successful ping test confirms that the virtual machine can communicate with external servers over the newly configured VPC and firewall.

Findings

VPCs and firewall rules are essential for cloud networking, providing security, traffic control, and flexibility. By using VPC subnets and firewall rules, we can create secure, scalable networks that protect data and control access. Firewalls ensure that only authorized traffic reaches important systems, keeping both security and performance strong. I successfully set up a Virtual Private Cloud (VPC) and configured firewall rules, enabling my VM to communicate effectively with the internet. This experience highlighted the importance of VPCs in segmenting resources and managing network traffic.

But, I faced some challenges during this process. Initially, configuring the VPC subnets and firewall rules was a bit overwhelming, as I had to ensure that the correct ports were open for both internal and external traffic. This experience reinforced the significance of networking in cloud environments. It highlighted how VPCs and firewall rules contribute to the security and performance of applications.

Conclusion

In conclusion I gained hands-on experience with Google Cloud services, such as cloud storage, virtual private cloud networking, and virtual machines, thanks to this assignment. I got knowledge about installing web servers, building up virtual machines, and serving content. Additionally, I learned how to manage access, set up automatic file deletion, and create Cloud Storage buckets. For safe networking, I also set up VPC and firewall rules.

These abilities are useful in the real world because they let businesses manage data, execute apps, and safeguard sensitive information. Businesses can create cost-effective, scalable, and flexible solutions that meet their needs by utilizing these Google Cloud services. Building efficient cloud infrastructures requires IT professionals to have a solid understanding of these tools.

References

- Google Cloud. *Compute Engine documentation*. Retrieved from https://cloud.google.com/compute/docs
- Google Cloud. *Cloud Storage documentation*. Retrieved from https://cloud.google.com/storage/docs
- Google Cloud. *Virtual Private Cloud (VPC) documentation*. Retrieved from https://cloud.google.com/vpc/docs
- Google Cloud. *Quickstart: Set up a VM instance*. Retrieved from https://cloud.google.com/compute/docs/quickstart-linux
- Google Cloud. *Creating and managing buckets*. Retrieved from https://cloud.google.com/storage/docs/creating-buckets
- Google Cloud. *VPC networking overview*. Retrieved from https://cloud.google.com/networking/docs/concepts/vpc
- Google Cloud. *Lifecycle management for Cloud Storage*. Retrieved from https://cloud.google.com/storage/docs/lifecycle