Google Cloud Study Jam – Day

Hands-on Lab: Compute Engine and NGINX Deployment

Overview

This lab was part of my **Google Cloud Study Jams** learning series.

The objective was to understand how to create and manage **Virtual Machines (VMs)** on Google Cloud using **Compute Engine**, and deploy a basic **NGINX web server** to host a simple web page.

This was my first step into learning **cloud infrastructure**, **VM management**, and **web server deployment** in Google Cloud Platform (GCP).

Lab Objectives

By completing this lab, I learned how to:

1. Create a virtual machine using the **Google Cloud Console**.
2. Create another VM using the **gcloud CLI** in **Cloud Shell**.
3. Install and configure an **NGINX web server** on the VM.
4. Understand **regions**, **zones**, and **firewall rules** in GCP.
5. Access a running instance through **SSH** and manage it remotely.

Tasks and Steps

1. Understanding Compute Engine

Compute Engine allows users to create virtual machines running on Google's infrastructure.

Each VM can run different operating systems like **Debian**, **Ubuntu**, or **Windows Server**, and can be scaled according to project needs.

2. Setting Up the Environment**

- Activated **Cloud Shell**, which provides a Linux terminal with pre-installed Google Cloud tools.

- Verified authentication and active project using:

```bash
 gcloud auth list
 gcloud config list project
```

Set default region and zone for the project:

bash

Copy code

```
gcloud config set compute/region europe-west1
export REGION=europe-west1
export ZONE=europe-west1-d
```

3. Creating the First VM (Console Method)

Navigated to: Compute Engine → VM Instances → Create Instance

Configuration:

Name: gcelab

Region: europe-west1

Zone: europe-west1-d

Machine Type: e2-medium (2 vCPUs, 4 GB RAM)

OS: Debian GNU/Linux 12 (bookworm)

Disk Size: 10 GB (Balanced Persistent Disk)

Firewall: Allowed HTTP traffic

Clicked Create, and after initialization, the VM was live and ready.

4. Installing NGINX on the VM

Connected to the VM using SSH directly from the console and ran:

bash

Copy code

```
sudo apt-get update
sudo apt-get install -y nginx
```

To verify NGINX is running:

bash

Copy code

```
ps auwx | grep nginx
```

Visited the External IP address of the VM in a browser, and saw the default:

“Welcome to NGINX!”

This confirmed the web server was successfully deployed.

5. Creating a VM Using gcloud CLI

Created another instance using the Cloud Shell command line:

bash

Copy code

```
gcloud compute instances create gcelab2 --machine-type e2-medium --zone=$ZONE
```

Verified creation:

bash

Copy code

```
gcloud compute instances list
```

Connected via SSH:

bash

Copy code

```
gcloud compute ssh gcelab2 --zone=europe-west1-d
```

Both instances (gcelab and gcelab2) appeared in the Compute Engine dashboard.

Real-World Example

Imagine you want to host your personal portfolio website or deploy a Flask ML model API —

Compute Engine lets you rent a cloud computer, set it up with NGINX or Flask, and make it publicly accessible by configuring firewall rules and external IPs.

In this lab, I essentially created:

A virtual server (the VM)

Installed a web server (NGINX)

Opened port 80 (HTTP) for public access

Key Concepts Learned

Concept	Description
Compute Engine	Virtual machines running on Google Cloud infrastructure
Cloud Shell	In-browser terminal with gcloud CLI pre-installed
Regions & Zones	Physical locations of Google’s data centers
Firewall Rules	Control inbound/outbound traffic (enabled HTTP access)
SSH	Secure method to connect to VMs remotely
NGINXWeb server	used to host and serve web pages

Outcome

Successfully deployed a VM-based web server on Google Cloud using both the Console and the CLI.

This lab helped me understand how cloud infrastructure works behind the scenes and how real-world web applications are deployed on the cloud.