# Creating a VM to Leverage Auto Scaling and Security in GCP

# Introduction

## Objective:

To set up a Virtual Machine (VM) instance in Google Cloud Platform (GCP), implement auto-scaling policies based on CPU utilization, and configure security measures including firewall rules and IAM roles. The project involves deploying a Compute Engine instance, creating a Managed Instance Group (MIG) with autoscaler enabled, defining ingress/egress firewall rules, and assigning IAM roles to restrict access. Additionally, the setup ensures high availability, scalability, and security by leveraging Cloud Load Balancing, Identity-Aware Proxy (IAP), and service accounts for controlled access.

**Virtual Machine Setup**

## Step 1: Instance Template Creation

## An Instance Template is required to create a Managed Instance Group (MIG) for auto-scaling. Before creating the template, we need to set up a custom VPC network and firewall rules for security.

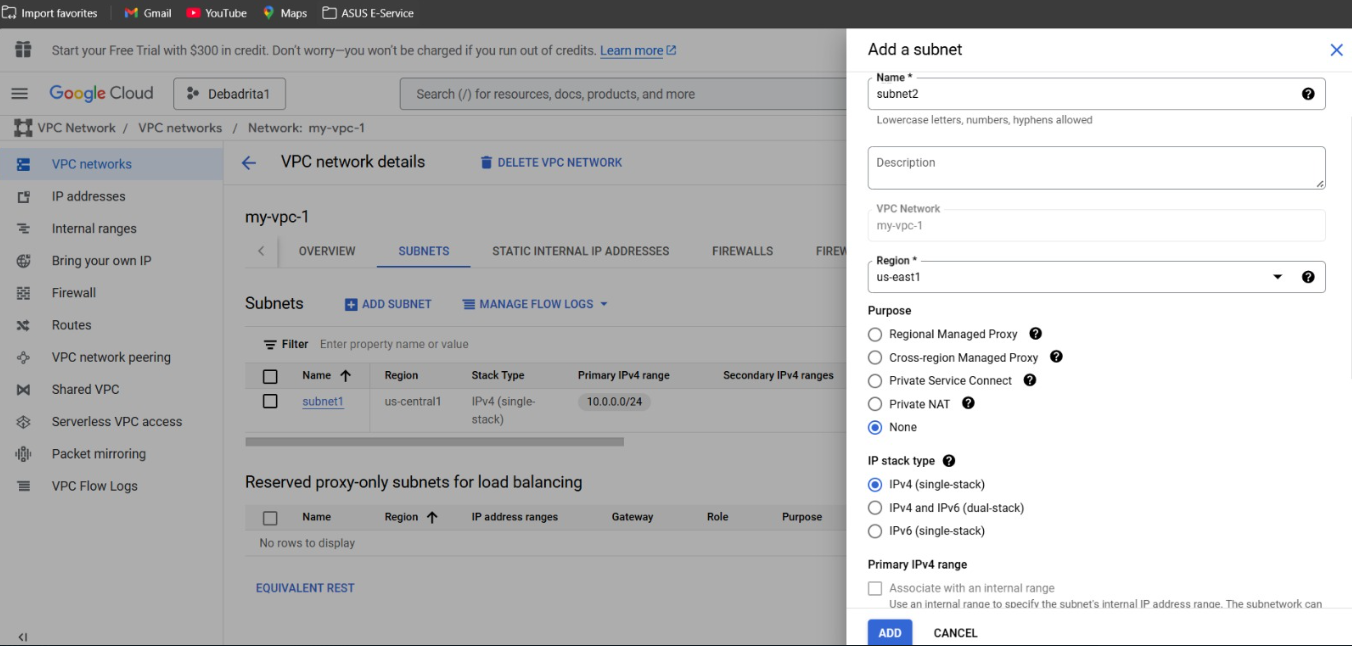
## 

**Creating a VPC Network:**

By default, GCP provides a **default network**, but for better control, we create a **custom VPC**.

#### Steps to Create a Custom VPC Network:

1. Go to the **Google Cloud Console** → Navigate to **VPC network** → Click **Create VPC network**.
2. Provide a **Name** (e.g., my-custom-vpc).
3. **Subnet Mode** → Select **Custom**.
4. Click **Add Subnet**:
   1. **Name**: subnet1
   2. **Region**: Select the desired region (e.g., us-central1)
   3. **IP range**: 10.0.0.0/24
   4. **Private Google Access**: Enable if needed.
   5. **Name**: subnet2
   6. **Region**: Select the desired region (e.g., us-central1)
   7. **IP range**: 10.0.0.1/24
   8. **Private Google Access**: Enable if needed.
5. Click **Create**.



## Next I have provided the necessary information to create the instance template including Location, Machine type and other necessary information provided nelow.

## 

## A white text box with black lines AI-generated content may be incorrect.

### Step 2: Creating a VM Instance (VM1) in Compute Engine

1. A standalone **VM instance (VM1)** can be created in **Google Cloud Compute Engine** using the steps below.

### ****Navigate to Compute Engine****

1. Open the **Google Cloud Console**.
2. Go to **Compute Engine** → Click **VM instances**.
3. Click **Create Instance**.

### ****Configure VM Instance (VM1)****

#### ****Basic Configuration****

1. **Name**: vm1
2. **Region**: Select the preferred region (e.g., us-central1).
3. **Zone**: Select a zone (e.g., us-central1-a).

#### ****Machine Configuration****

1. Machine Series: E2
2. Machine Type: e2-medium (2 vCPUs, 4 GB RAM) or as per requirement.

#### ****Boot Disk****

1. Click **Change** to select an OS:
2. Choose **Ubuntu 22.04 LTS** (or your preferred OS).
3. Click **Select**.

#### ****Networking****

1. Expand the Advanced Options.
2. Under Network Interfaces:
   * Network: Select my-vpc-1
   * Subnetwork: Select subnet1.

#### ****Firewall****

1. **Check** Allow **HTTP** and **SSH** traffic.

**Step 3: Configuring Firewall Rules and IAM for VM1**

**Creating a Firewall Rule in GCP**

**Steps to Allow Traffic to VM1**

1. Open Google Cloud Console.
2. Navigate to VPC Network → Click Firewall.
3. Click Create Firewall Rule.

**Configure the Firewall Rule**

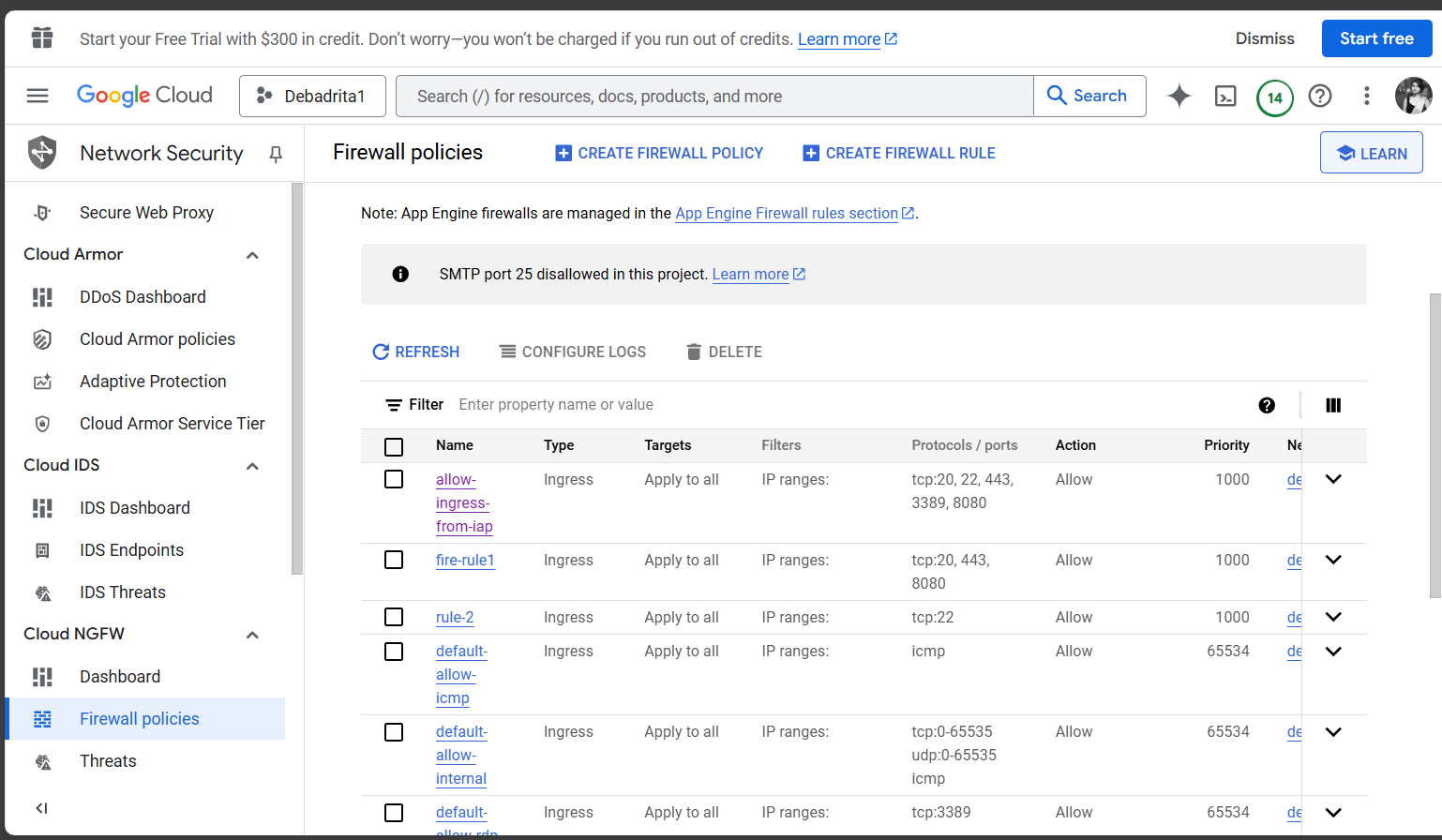
1. Name: allow-ingress-from-iap
2. Network: Select my-custom-vpc (or default if no VPC is created).
3. Direction: Ingress (incoming traffic).
4. Action: Allow.
5. Targets:
6. Select Specified target tags.
7. Enter vm1-tag (or any tag assigned to VM1).
8. Source filter: IPv4 Ranges.
9. Source IP Ranges:
10. [0.0.0.0/0](http://0.0.0.0/0) (Allow access from anywhere, modify for security).
11. Protocols and ports:
12. TCP: Enter 22, 80 (for SSH and HTTP access).

**Apply the Firewall Rule**

1. Click **Create**.

A screenshot of a computer

AI-generated content may be incorrect.



**Setting Up IAM Roles for VM1**

**Grant IAM Permissions**

1. Open Google Cloud Console.
2. Navigate to IAM & Admin → Click IAM.
3. Click Grant Access.

**Assign IAM Roles**

1. Principal: Enter the Google Account or Service Account.
2. Role: Select a suitable role based on access needs:
3. Compute Admin: Full access to manage VM instances.
4. Compute Viewer: Read-only access.
5. Custom Role: If specific permissions are needed.
6. Click Save.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Creating a Managed Instance Group (MIG) in GCP**

A Managed Instance Group (MIG) ensures auto-scaling and high availability. We'll create instance-group-2 with 5 instances, based on template-2.

**Navigate to Instance Groups**

1. Open Google Cloud Console.
2. Go to Compute Engine → Click Instance groups.
3. Click Create Instance Group.

**Configure the Instance Group**

**Basic Settings**

1. Name: instance-group-2
2. Location:
3. Regional: Spread across multiple zones (recommended).
4. Region: us-central1
5. Zones: us-central1-c (or multiple for high availability).

**Instance Template**

1. Instance Template: Select template-2.

**Autoscaling Configuration**

1. Autoscaling: On
2. Target CPU Utilization: 60%
   1. Instances scale up when CPU usage > 60%.
   2. Instances scale down when CPU usage < 60%.
3. Minimum Instances: 2
4. Maximum Instances: 6

**Health Check (Optional)**

1. Create a health check (Recommended for auto-restarts).
2. Protocol: HTTP
3. Port: 80
4. Check Interval: 10s
   * Timeout: 5s
   * Unhealthy Threshold: 3
   * Healthy Threshold: 2

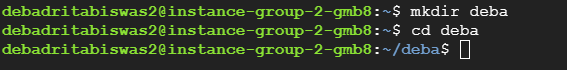
**Create the Instance Group**

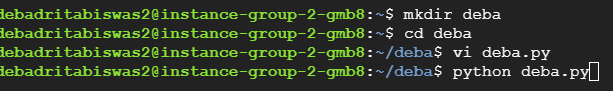
1. Click Create.

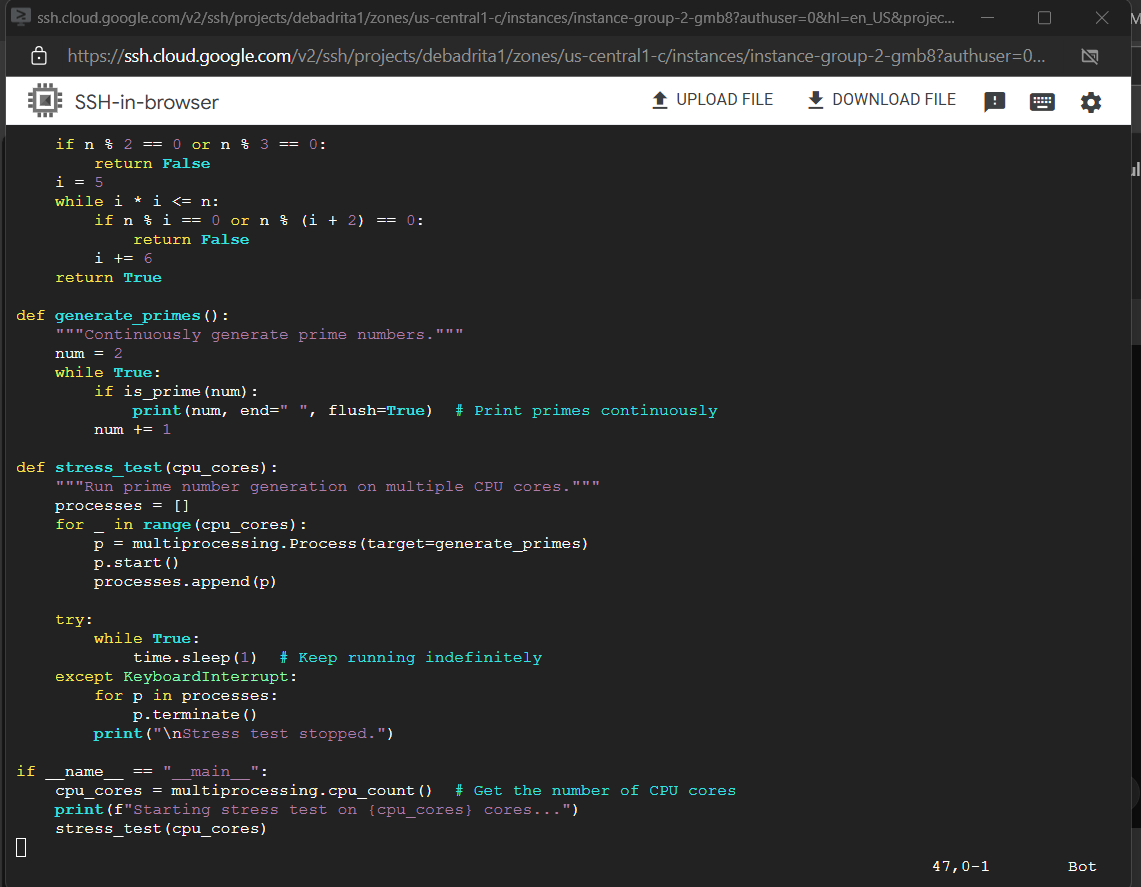
A screenshot of a computer

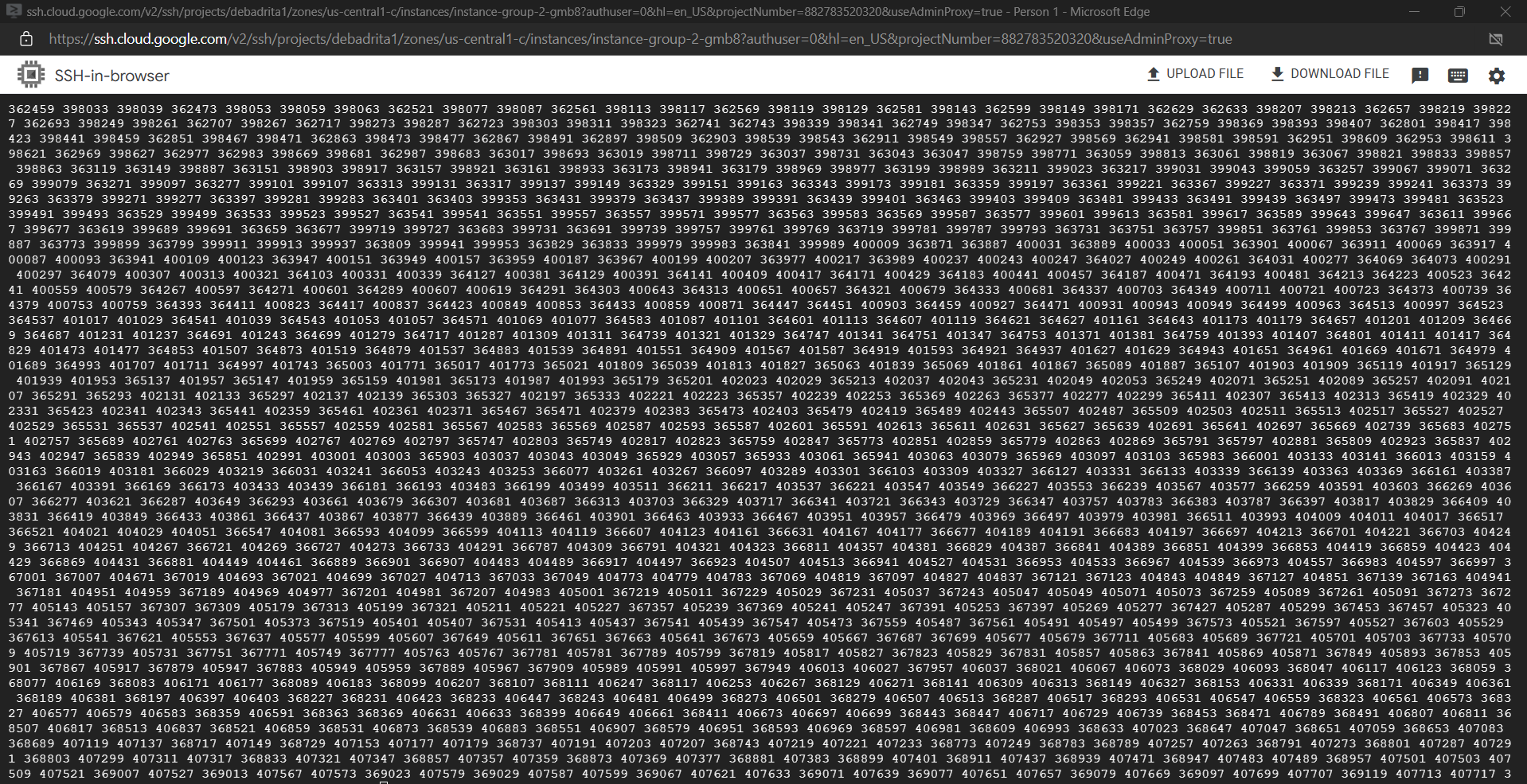
AI-generated content may be incorrect.

**I ran a program which would run infinitely and thus saw once the 2 VMs were exhausted by CPU utilization of 60%, new VMs were automatically getting created.**









Thus, we can see multiple instance is getting created as the CPU load has crossed 60 percent in both the machines.

A screenshot of a computer

AI-generated content may be incorrect.

Checking CPU utilization by monitoring:

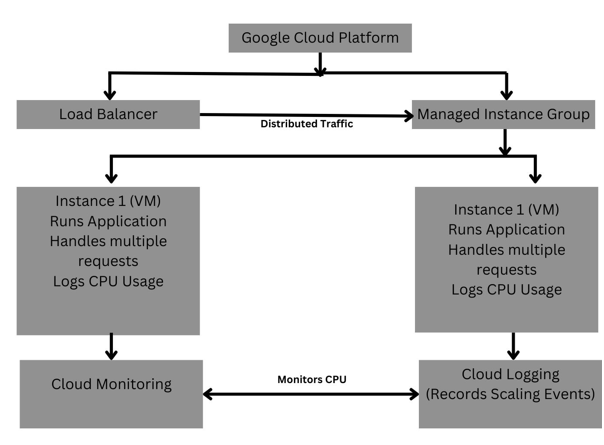
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Architecture Diagram**

****

**Autoscaling:**

**A white background with blue text

AI-generated content may be incorrect.**

1. **Source Code Repository**

## Github Repository Link: [debadrita11/GCP\_VM\_Creation](https://github.com/debadrita11/GCP_VM_Creation)

# Recorded Video Demo

1 st part:

https://drive.google.com/file/d/1BeEq8xZ\_M7yrlshwmtLxzaZHXSdhY-br/view?usp=gmail

2nd Part:

https://drive.google.com/file/d/1Vsa0BI2Uic-XcorNbGI-3haFhdi2dWUO/view?usp=drive\_web

# 4. Conclusion

In this assignment, we successfully set up a virtual machine (VM) in Google Cloud Platform (GCP), implemented auto-scaling policies based on CPU utilization, and configured security measures such as firewall rules and IAM roles. The auto-scaling mechanism dynamically adjusted the number of VM instances in response to workload fluctuations, ensuring optimal resource utilization and performance efficiency.

By creating an instance template, defining a managed instance group, and applying auto-scaling policies, we observed how instances scaled up and down based on CPU load. Additionally, security measures were enforced by restricting access through IAM roles and configuring firewall rules to control network traffic.

This implementation demonstrates the effectiveness of cloud-based auto-scaling in handling workload variations while maintaining security and cost efficiency. It also highlights the importance of automation in modern cloud environments, ensuring scalability, availability, and security in cloud-based applications.