Technical Report :Automating Scalable Cloud Infrastructures on Google Cloud Platform

1. Introduction

This report documents the deployment process of a **Managed Instance Group (MIG)** on **Google Cloud Platform (GCP)** using an automated **Bash script**, with a primary focus on **auto-scaling**. The infrastructure is designed to dynamically adjust the number of virtual machine instances based on workload demand, ensuring optimal performance and cost efficiency.

Key Objectives:

- **Implementing auto-scaling policies** to increase or decrease instances based on **CPU utilization thresholds**, ensuring high availability while reducing unnecessary resource consumption.
- **Automating virtual machine provisioning** using an **Instance Template** for consistency and scalability.
- **Configuring security measures**, including:
  - **IAM roles** to manage access control.
  - **Firewall rules** to regulate network traffic and protect cloud resources.

By automating the **deployment and scaling processes**, the infrastructure efficiently handles variable workloads, reducing manual intervention and ensuring a reliable, cost-effective cloud environment.

2. Architecture Diagram:

The architecture represents an auto-scaling infrastructure on Google Cloud Platform (GCP), ensuring scalability, security, and controlled access.

Key Components:

Auto-Scaling Group – Manages VM instances dynamically based on CPU utilization.

Virtual Machines (VMs) – Runs Apache web servers, created using an instance template.

Cloud Firewall Rules –

allow-ssh-my-ip → Restricts SSH access to a specific IP.

allow-http → Enables HTTP traffic on port 80.

allow-health-check → Allows Google Cloud health checks.

IAM Roles –

Viewer Role: Read-only access to compute resources.

Admin Role: Grants full control over instances.

Network Communication – Users access VMs via SSH or HTTP through firewall and IAM-controlled policies.

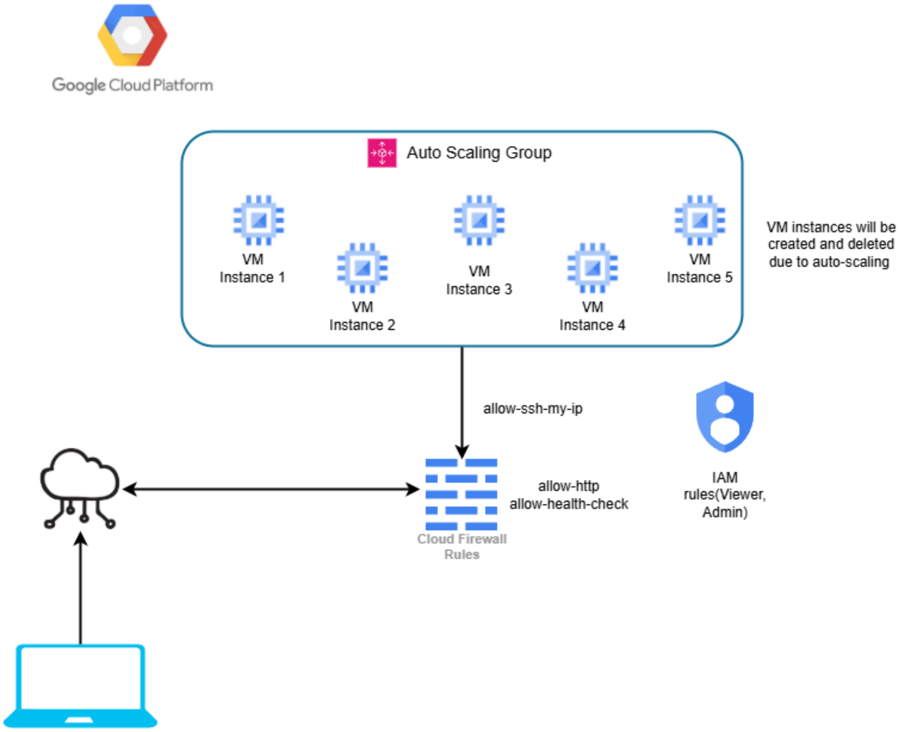


Fig: Auto-Scaling Architecture on GCP

3. Pre-Checks and Setup

System Requirements

To ensure a successful execution, the script first verifies the following requirements:

- **Non-root execution**: Running the script as root is restricted to prevent permission conflicts.
- **Dependency check**: Ensures the availability of curl and sudo. If missing, the script installs them.

Google Cloud SDK Installation

If **Google Cloud SDK** is not detected, the script:

- Adds the official Google Cloud repository
- Installs the SDK
- Configures authentication

```
gcloud config set project "$PROJECT_ID"

gcloud config set compute/region "$REGION"

gcloud config set compute/zone "$ZONE"
```

Authentication

Before executing commands, authentication is verified. If no authenticated user is found, a login prompt appears.

4. Deploying Cloud Infrastructure

Creating an Instance Template

An **instance template** is created, defining the VM configuration:

- **Machine Type:** e2-micro
- **Operating System:** Ubuntu 20.04 LTS
- **Startup Script:** Installs Apache and serves a default webpage

Creating a Managed Instance Group

A **Managed Instance Group (MIG)** is deployed using the instance template with a minimum of **1 instance** and a maximum of **5 instances**.

```
gcloud compute instance-groups managed create "$INSTANCE_GROUP" \
  --base-instance-name=web-instance \
  --template="$INSTANCE_TEMPLATE" \
  --size="$MIN_REPLICAS" \
  --zone="$ZONE"
```

Enabling Auto-Scaling

The auto-scaling policy is configured as follows:

- **Target CPU utilization:** 60%
- **Cool-down period:** 120 seconds

5. Security and Access Control

Firewall Configuration

Firewall rules are set up to:

- Allow HTTP traffic on port **80**
- Restrict SSH access to a specific IP

```
gcloud compute firewall-rules create allow-http \
  --allow=tcp:80 \
  --source-ranges=0.0.0.0/0 \
  --target-tags=http-server
```

IAM Role Assignment

Appropriate **IAM roles** are assigned to grant specific permissions:

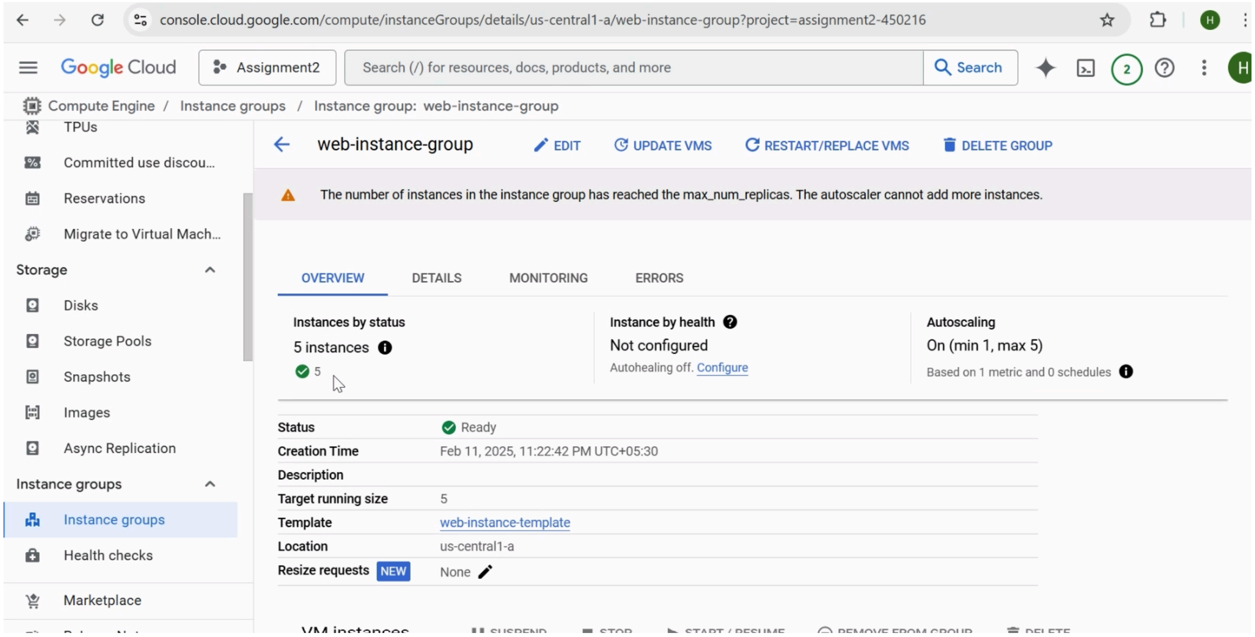
- **Compute Viewer Role** to er.himani1998@gmail.com
- **Compute Instance Admin Role** to m23csa516@iitj.ac.in

6. Testing:

Auto-Scaling Verification

A simulated high CPU workload is used to trigger the auto-scaler, ensuring additional instances are created dynamically.

```
stress --cpu 4 --timeout 100s
```



The number of instances in the instance group has reached the max_num_replicas. The autoscaler cannot add more instances.

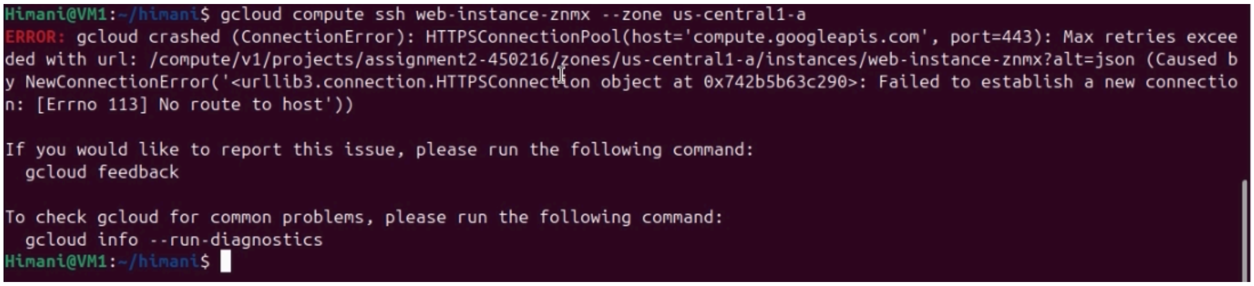
Instances by status	Instance by health	Autoscaling
5 instances ✓ 5	Not configured Autohealing off. Configure	On (min 1, max 5) Based on 1 metric and 0 schedules

Status	✓ Ready
Creation Time	Feb 11, 2025, 11:22:42 PM UTC+05:30
Description	
Target running size	5
Template	web-instance-template
Location	us-central1-a
Resize requests NEW	None

Fig: Auto-Scaling with 5 instances on GCP

Checking firewall Rules

Since the error occurred **after switching networks**, it's likely that the new network **does not have the required firewall rules** allowing SSH (tcp:22) or HTTP (tcp:80) traffic. The firewall rules are tied to a **specific IP network**, so changing the network requires reconfiguring security policies.

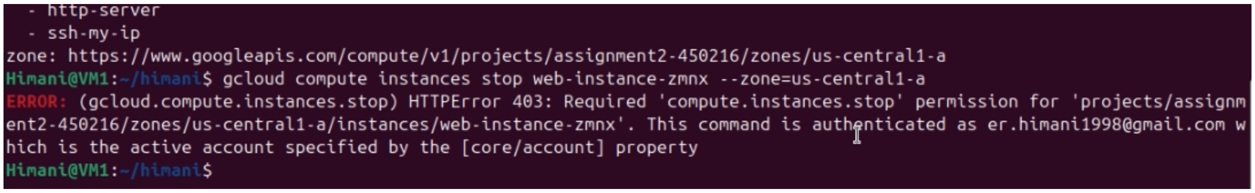


```
Himani@VM1:~/himani$ gcloud compute ssh web-instance-znmx --zone us-central1-a
ERROR: gcloud crashed (ConnectionError): HTTPSConnectionPool(host='compute.googleapis.com', port=443): Max retries excee
ded with url: /compute/v1/projects/assignment2-450216/zones/us-central1-a/instances/web-instance-znmx?alt=json (Caused b
y NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x742b5b63c290>: Failed to establish a new connectio
n: [Errno 113] No route to host'))

If you would like to report this issue, please run the following command:
  gcloud feedback

To check gcloud for common problems, please run the following command:
  gcloud info --run-diagnostics
Himani@VM1:~/himani$ 
```

Fig: Firewall rules testing

Checking IAM Roles

The user er.himani1998@gmail.com have ‘viewer’ role and the user m23csa516@iitj.ac.in have owner and ‘Instance Admin’ role.



```
  - http-server
  - ssh-my-ip
zone: https://www.googleapis.com/compute/v1/projects/assignment2-450216/zones/us-central1-a
Himani@VM1:~/himani$ gcloud compute instances stop web-instance-zmnx --zone=us-central1-a
ERROR: (gcloud.compute.instances.stop) HTTPError 403: Required 'compute.instances.stop' permission for 'projects/assignm
ent2-450216/zones/us-central1-a/instances/web-instance-zmnx'. This command is authenticated as er.himani1998@gmail.com w
hich is the active account specified by the [core/account] property
Himani@VM1:~/himani$
```

Fig: IAM roles testing

Stopping of VM failed as the user doesn’t have admin privilege. The user logged in has only viewing permission. Thus, IAM roles are working correctly.

7. Conclusion:

The deployment of a Managed Instance Group (MIG) with auto-scaling policies on Google Cloud Platform (GCP) successfully ensures scalability, security, and efficiency. The infrastructure dynamically adjusts VM instances based on CPU utilization, optimizing resource usage while maintaining high availability. Firewall rules safeguard access by restricting SSH and allowing only essential traffic, while IAM roles enforce controlled permissions. The automated deployment script streamlines provisioning, reducing manual effort and ensuring consistency. This setup provides a reliable, secure, and cost-effective cloud solution, with potential future enhancements such as load balancing and advanced monitoring to further optimize performance.

Github Link: https://github.com/m23csa516/VCC_Assignment2.git

Google Drive Link of video:
https://drive.google.com/file/d/1Xzw1O-_5JhlpSjXX4M7cD42Hb70ZaTxV/view?usp=sharing