# **Topic: Report on GCP VM with Auto-Scaling and Security Implementation**

**Contributor:**



Reda Fatima Ali (Indian Institute of Technology, Jodhpur)

Roll: M23AID008

Department: Data and Computational Sciences

Subject Code: CSL7510

**Contents**

[**Introduction : Report on GCP VM with Auto-Scaling and Security Implementation** 1](#_Toc191925500)

[**Architecture Design** 3](#_Toc191925501)

[**Creation of VM instance on GCP** 4](#_Toc191925502)

[**Configuration of Auto-Scaling Policies on GCP** 5](#_Toc191925503)

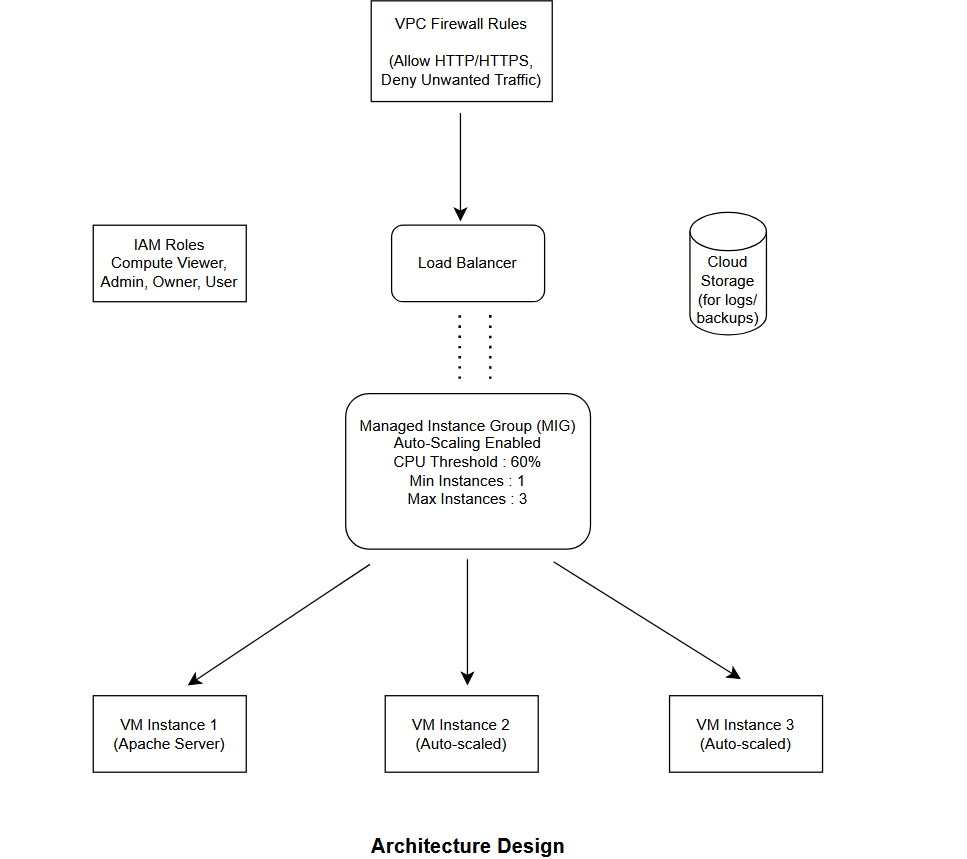
[**Implementation of Security Measures** 7](#_Toc191925504)

[**Setting up IAM Roles for Restricted Access** 7](#_Toc191925505)

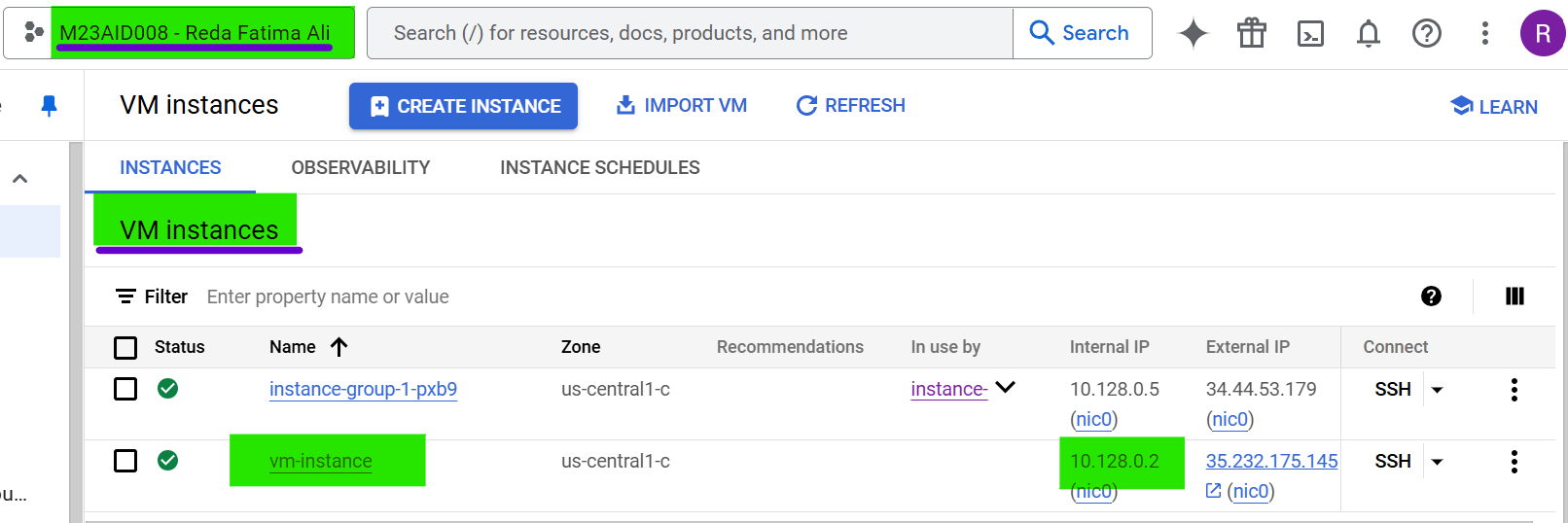
[**Configuration of firewall rules to allow/deny traffic** 8](#_Toc191925506)

[**Load Testing (Stress) - To verify the Autoscaling** 9](#_Toc191925507)

# **Architecture Design**



# **Creation of VM instance on GCP**

* Go to Google Cloud Console: <https://console.cloud.google.com/>
* Navigate to Compute Engine > VM Instances.
* Click 'Create Instance'.
* Enter the instance name (“*vm-instance*”).
* Choose a region (e.g., us-central1) and zone.
* Select the machine type (e2-micro for free tier).
* Select the boot disk (Debian/Ubuntu).
* Under Firewall, check 'Allow HTTP traffic' and 'Allow HTTPS traffic' if needed.
* Click 'Create' to launch the VM.

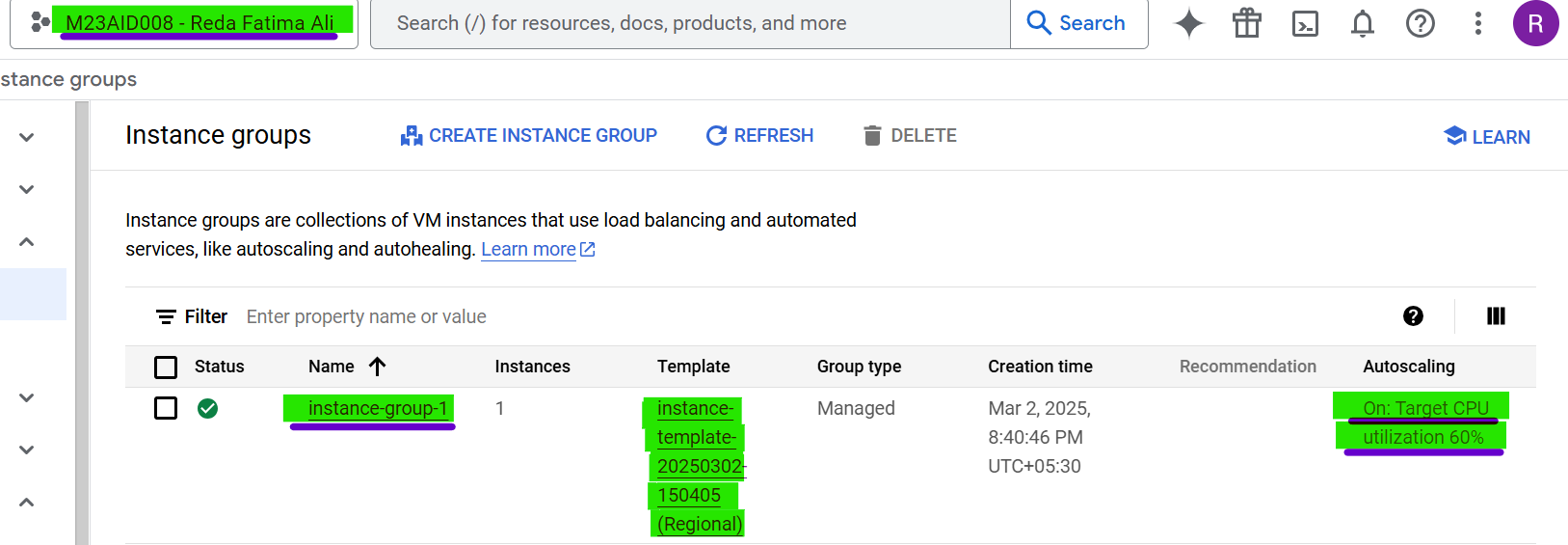
**Fig. 1:** VM Instance Created (“*vm-instance*”)

Reasons to go for the above steps w.r.t creation of VM instance on GCP

* We start by creating a VM as the base compute resource needed to run applications and simulate workloads.
* Creating a VM in GCP provides a scalable, secure, and highly available compute environment with minimal setup.
* Choosing the free-tier eligible machine type (e2-micro) helps manage costs effectively while testing.

# **Configuration of Auto-Scaling Policies on GCP**

* Navigate to Compute Engine > Instance Groups.
* Click 'Create Instance Group'.
* Name your group (e.g., my-instance-group).
* Choose the location (same as the VM region).
* Select an Instance Template (create one if needed with your VM configuration).
* Enable Autoscaling (with below configurations):
  + Mode: On
  + Metric: CPU Utilization
  + Target CPU Utilization: 60%
  + Minimum Instances: 1
  + Maximum Instances: 3
* Click 'Create' to set up the instance group with auto-scaling enabled.



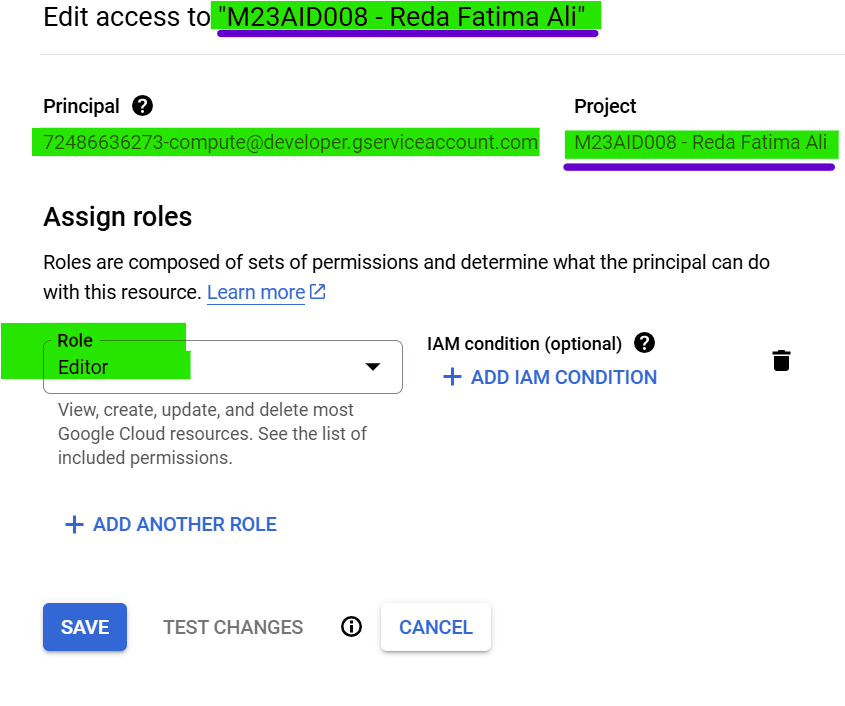
**Fig. 2:** Instance Group Created & Configuration of auto scaling (“*instance-group-1*”)

Reasons to go for the configuration of auto scaling policy on GCP

* Auto-scaling is configured to handle unpredictable traffic spikes, ensuring the application remains responsive.
* Auto-scaling automatically adjusts the number of VM instances based on workload, ensuring optimal performance.
* Helps control costs by scaling down resources during low usage periods.

# **Implementation of Security Measures**

## **Setting up IAM Roles for Restricted Access**

* Navigate to IAM & Admin > IAM.
* Click 'Grant Access' or 'Add'
* Enter the email address of the user.
* Assign the roles.
* Click 'Save' to apply the roles.

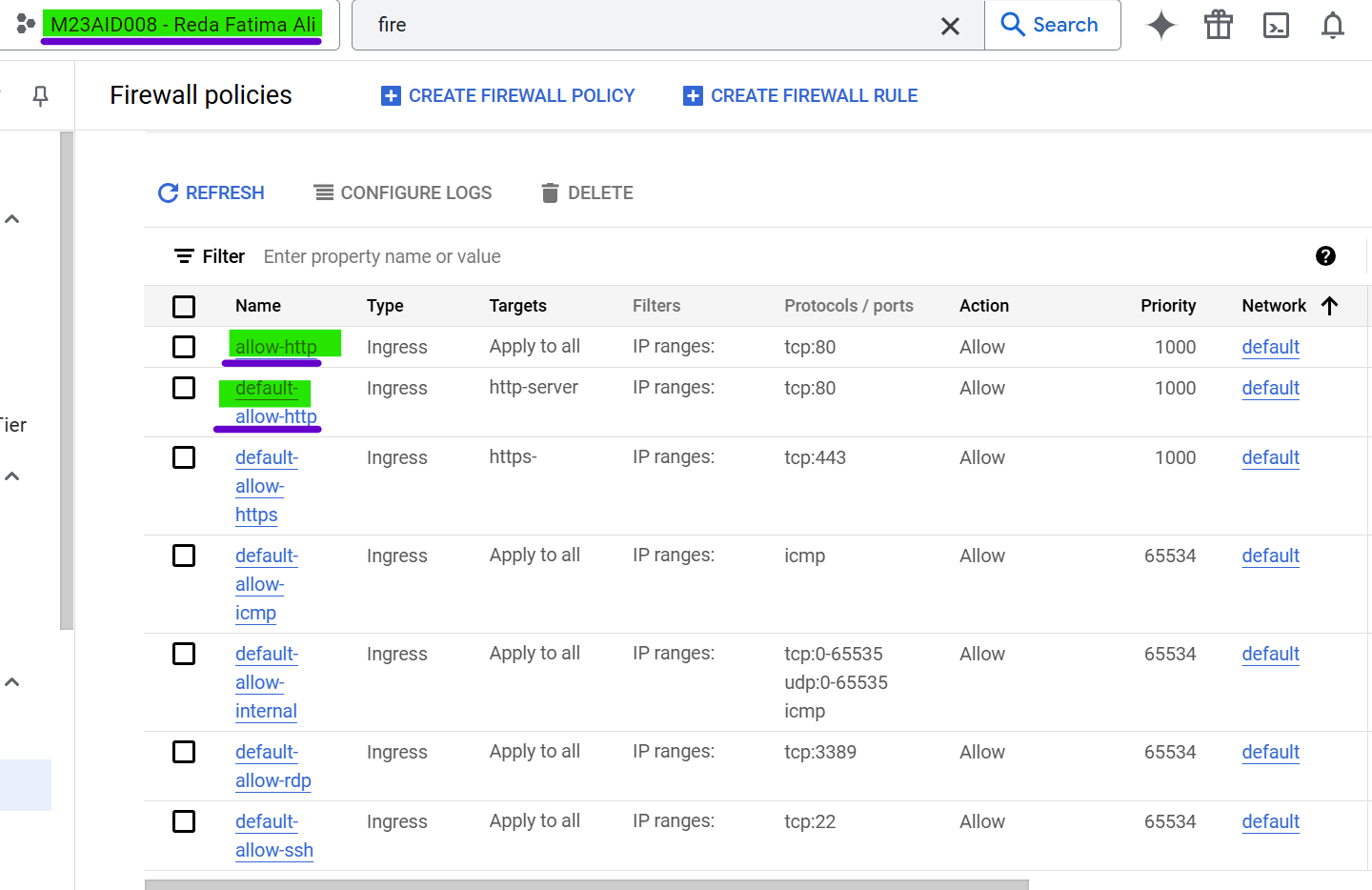
**Fig. 3:** IAM Roles

Reasons to have IAM Roles Implemented

* Securing the environment is critical; setting up IAM prevents unauthorized access to sensitive resources.
* IAM ensures only authorized users have access, reducing the risk of accidental or malicious actions.
* Roles provide fine-grained access control, adhering to the principle of least privilege.

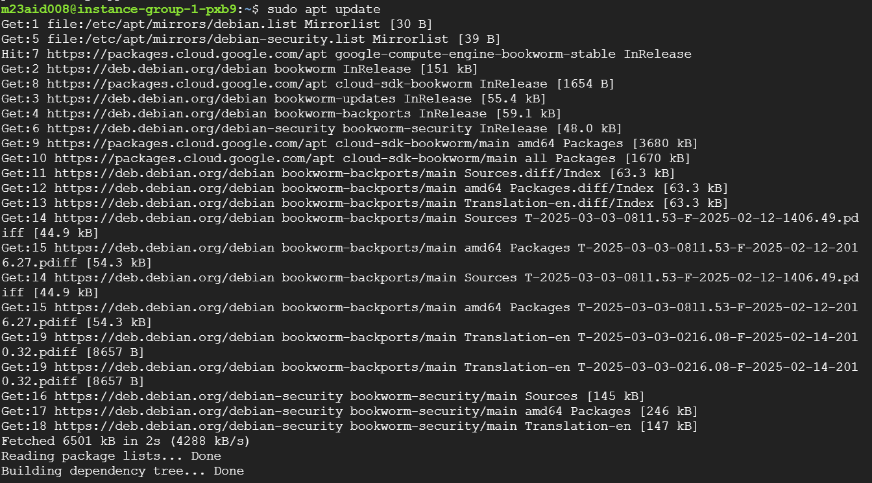
## **Configuration of firewall rules to allow/deny traffic**

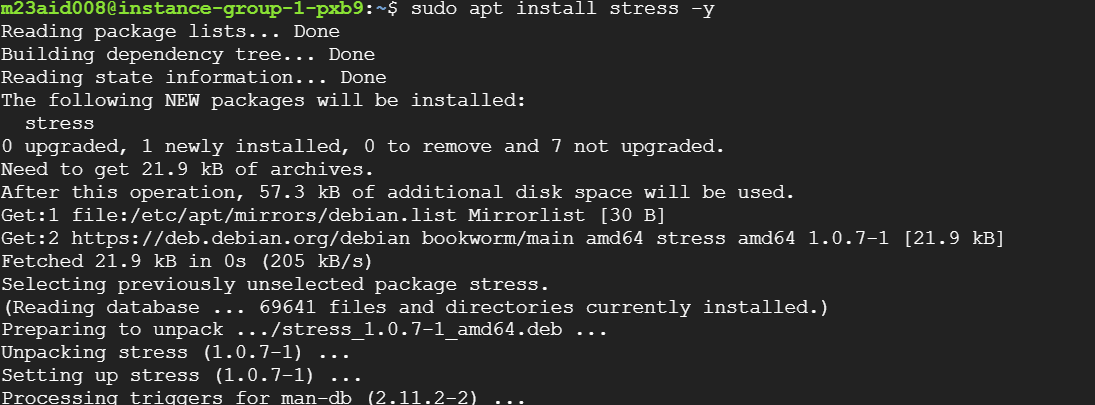
* Navigate to VPC Network > Firewall Rules.
* Click **'Create Firewall Rule'**.
* Enter a name (e.g., allow-http).
* Direction: Ingress.
* Targets: All instances in the network or specific tags.
* Source IP ranges: 0.0.0.0/0 (for public access).
* Protocols and ports: TCP: 80 (HTTP), TCP: 443 (HTTPS).
* Click 'Create' to apply the rule.



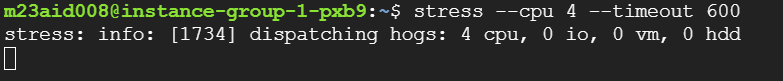
**Fig. 4:** Firewall Rules

# **Load Testing (Stress) - To verify the Autoscaling**

* Step 1: **SSH into the instance group**, created above
* Step 2: **sudo apt update**
* Step 3: **Install Stress** – **sudo apt install stress -y**



* Step 4: **Run stress to max CPU** – **stress –cpu 4 –timeout 600**



* While Load is running, **watch for new instances getting created automatically as CPU usage exceeds the auto-scaling threshold**.

