Assignment3: Auto-Scaling Local VM to Google Cloud Platform (GCP)

1. Introduction

The objective of this assignment is to create a **local virtual machine (VM)** and implement a mechanism to **monitor resource usage** (CPU and/or memory). When resource utilization exceeds **75%**, additional compute resources are automatically **provisioned in a public cloud (GCP)**, ensuring seamless scaling and performance optimization without manual intervention.

Key Objectives

- **Local VM Creation**: Setup using VirtualBox.
- **Resource Monitoring**: Implemented with a monitoring script.
- **Auto-Scaling to Cloud**: Automate provisioning of additional resources on GCP.
- **Sample Application Deployment**: Demonstrate the entire flow using a simple web application.

2. Architecture Overview

This section presents an overview of the **auto-scaling architecture**, describing its key components and interactions.

2.1 High-Level Architecture

The system consists of the following major components:

1. **Local VM**
    - Runs **lighttpd web server** and a monitoring script.
    - Monitors **CPU and memory usage** continuously.
    - If **CPU usage exceeds 75%**, it triggers the auto-scaling process.
2. **Resource Monitoring**
    - A **custom Bash script** continuously monitors CPU and memory usage.
    - If the threshold is exceeded, the script initiates the **scaling process** on GCP.
3. **Google Cloud Platform (GCP) Components**

  - **Compute Engine**: Uses a **Managed Instance Group (MIG)** for auto-scaling.
  - **Instance Template**: Defines VM specifications for auto-scaling.
  - **Load Balancer**: Distributes traffic among auto-scaled instances.
  - **Cloud Firewall Rules**: Allows HTTP and health-check traffic.
  - **Cloud Storage (GCS)**: Stores web application files.

2.2 Auto-Scaling Flow

- The **local VM monitors** CPU usage.
- If **CPU exceeds 75%**, the script:
  - Uploads web content to **Cloud Storage**.
  - Creates **GCP VM instances** using **Managed Instance Groups (MIG)**.
  - Configures a **Load Balancer** to distribute traffic.
  - Updates the **local Apache configuration** to forward traffic to GCP.
- If CPU exceeds 75% again, **MIG size increases dynamically**.

3. Architecture Diagram

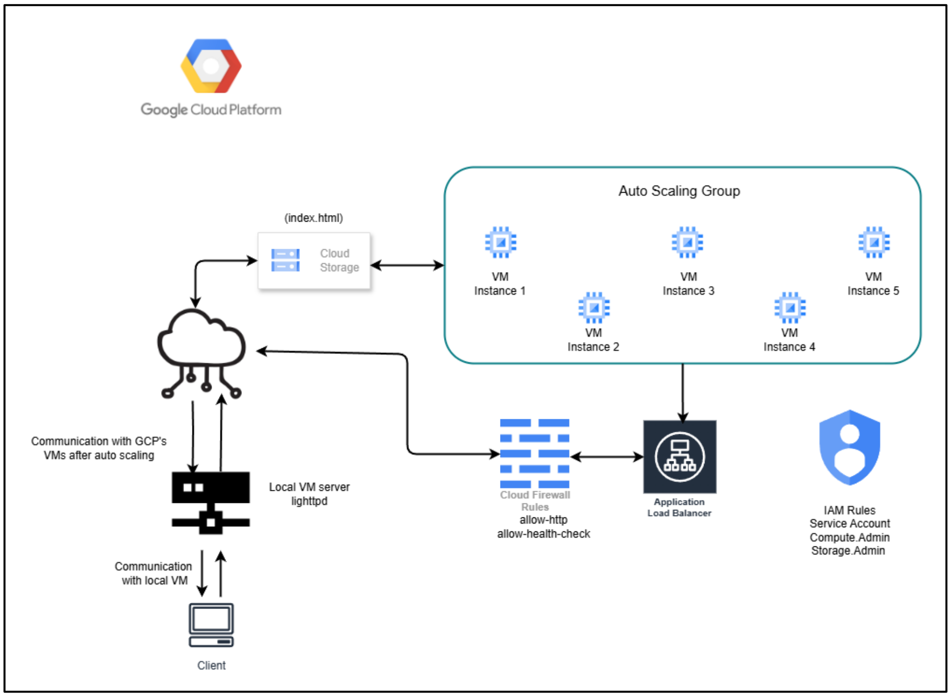

Fig: Architecture Diagram

Diagram Explanation

The diagram illustrates the **communication flow** between components:

- The **client** sends requests to the **local VM**.
- The **local VM** monitors resource usage.
- If CPU usage **exceeds 75%**, traffic is forwarded to **GCP instances**.
- A **Cloud Load Balancer** manages incoming requests to GCP VMs.
- **IAM roles** are assigned to manage security and resource provisioning.

4. Step-by-Step Implementation Guide

4.1 Prerequisites

Local Environment:

- A system with **VirtualBox or VMware** installed.
- Ubuntu or any **Linux-based guest OS** running inside the VM.

Google Cloud Platform Requirements:

- A **GCP account with billing enabled**.
- **Compute Engine API** and **IAM roles** to create instances.
- **Google Cloud SDK installed** locally.

4.2 Local VM Setup

1. **Install VirtualBox / VMware**.
2. **Create a new Ubuntu VM** with:
   - **2 vCPUs, 4GB RAM**.
   - Internet access enabled (**Bridged Adapter**).
3. **Install a web server (lighttpd)**:

```
sudo apt update -y

sudo apt install lighttpd -y
```

4.3 Configuring the Monitoring & Auto-Scaling Script

```
bash auto_scaling_script.sh
```

- Installs **Google Cloud SDK**.
- Creates **GCP Service Accounts & IAM roles**.
- Begins **resource monitoring.**

4.4 Auto-Scaling Actions on GCP

- The script **uploads local web content** to **Cloud Storage**.
- Creates an **Instance Template**.
- Deploys a **Managed Instance Group (MIG)**.
- Configures a **Cloud Load Balancer**.

5. Testing the Auto-Scaling Setup

1. **Run the script & monitor output:**

   ```
   bash /auto_scaling_script.sh
   ```

2. **Apply high CPU load using stress testing:**

   ```
   stress --cpu 4 --timeout 60s
   ```

3. **Observe auto-scaling behavior** in the GCP console.
4. **Verify requests being served** from GCP instances.

6. Source Code Repository

The source codes used for this implementation are available at the following repository:

- **GitHub**: https://github.com/m23csa516/VCC_Assignment3

7. Link to Recorded Video Demo

Here is a link to a recorded video : https://drive.google.com/file/d/1_B3MMx2oLUWpDltxzPvd2FP04SWDCdmz/view?usp=sharing, demonstrating the setup process, which shows the auto-scaling and security configurations.

References:

- https://cloud.google.com/sdk?hl=en
- https://cloud.google.com/cli?hl=en
- https://cloud.google.com/compute/docs/autoscaler
- https://cloud.google.com/firewall/docs/firewalls
- https://cloud.google.com/iam/docs/overview