

Google Cloud Platform (GCP) Virtual Machine Setup with Auto-Scaling and Security

1. Introduction

Google Cloud Platform (GCP) offers an extensive suite of products built for the cloud to enable businesses deploy, manage, and scale apps easily. Customers are able to scale and agile through one of this products - Compute Engine, which allows customers to provision and manage virtual machines (VMs). In cloud environments, dynamic auto-scaling optimally adjusts resources based on the existing workload to ensure maximum performance while minimizing costs. For cybersecurity purposes, resources are protected from unauthorized access and usage by IAM (Identity and Access Management) and firewall rules. This guide, provides a detailed guide on deploying an instance of VM and thoroughly explains how to configure automatic-scaling policies on GCP and other security settings to maintain scalability.

2. Objectives

The main objectives of this project are as follows:

- Deploy a Virtual Machine (VM) instance in GCP.
- Configure Auto-Scaling Policies to optimize resources dynamically based on workload demands.
- To implement Security Measures such as IAM roles and firewall rules for protecting cloud resources from unauthorized access.
- Provide a clear architectural design representing the system interactions.
- Include source code repositories for deployment automation.

3. Step-by-Step Implementation

3.1 Creating a VM Instance in GCP

The Virtual Machine (VM) is used to run programs in a cloud infrastructure. Following is the detailed process to launch a VM instance in Google Cloud Platform.

Step 1: Log in to GCP Console

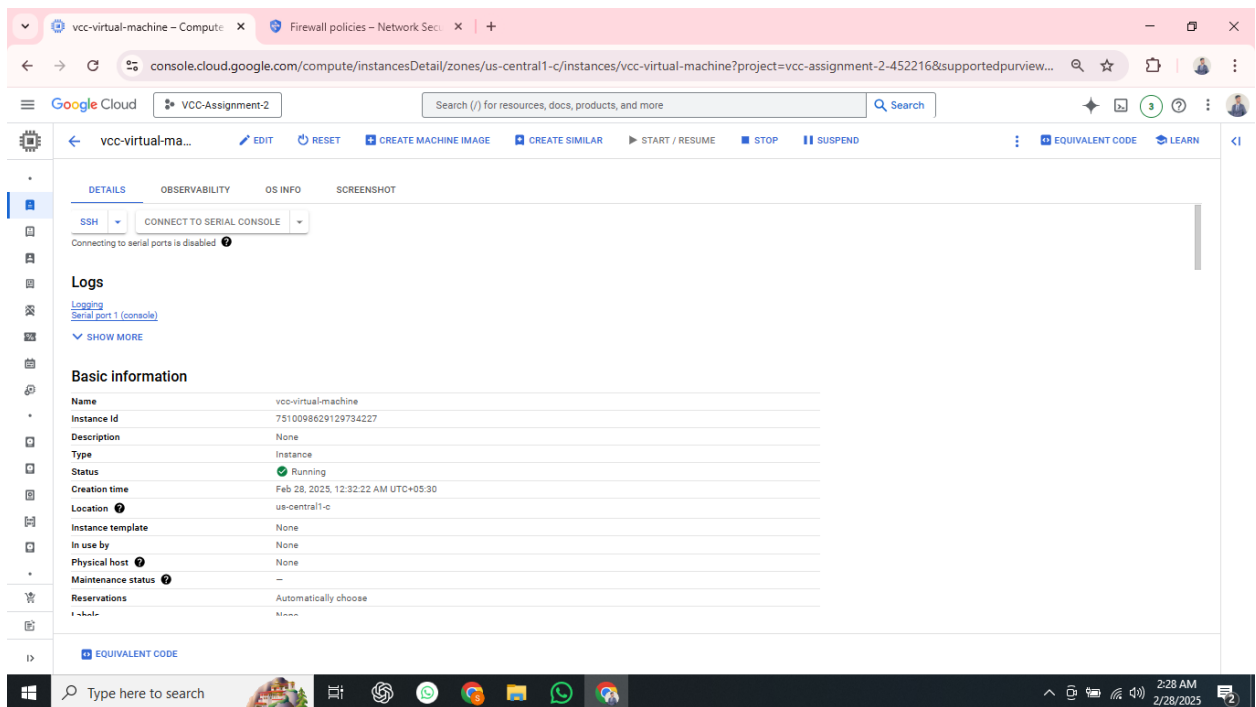
1. Open a web browser -> navigate to [Google Cloud Console](#).
2. Sign in with your Google account.

Step 2: Navigate to Compute Engine

1. In the left-hand navigation menu, go to Compute Engine > VM Instances.
2. Click Create Instance.

Step 3: Configure the VM Instance

1. **Enter Instance Name:** e.g., my-instance.
2. **Select a Region & Zone:** Choose a region nearest to your users for better performance.
3. **Choose a Machine Type:** Select e2-medium (2 vCPUs, 4 GB RAM) or adjust based on workload needs.
4. **Boot Disk Selection:**
 - Click Change and select Ubuntu 20.04 LTS.
5. **Enable Firewall:** Check the boxes for Allow HTTP and HTTPS traffic.
6. Click **Create** to launch the instance.



3.2 Configuring Auto-Scaling Policies

Auto-scaling ensures that additional VM instances are created when CPU utilization increases and reduces instances when demand is low, optimizing cost and performance.

Step 1: Create a Managed Instance Group

1. Navigate to Compute Engine > Instance Groups.
2. Click Create Instance Group.

Step 2: Configure Instance Group

1. Choose Managed Instance Group.
2. Select an Instance Template (Create one if necessary).
3. Set Auto-Scaling to Enabled.

Step 3: Define Auto-Scaling Policies

1. **Metric:** CPU utilization.

2. **Target CPU Utilization: 60%.**
3. **Minimum Instances: 1.**
4. **Maximum Instances: 5.**
5. Click **Create**.

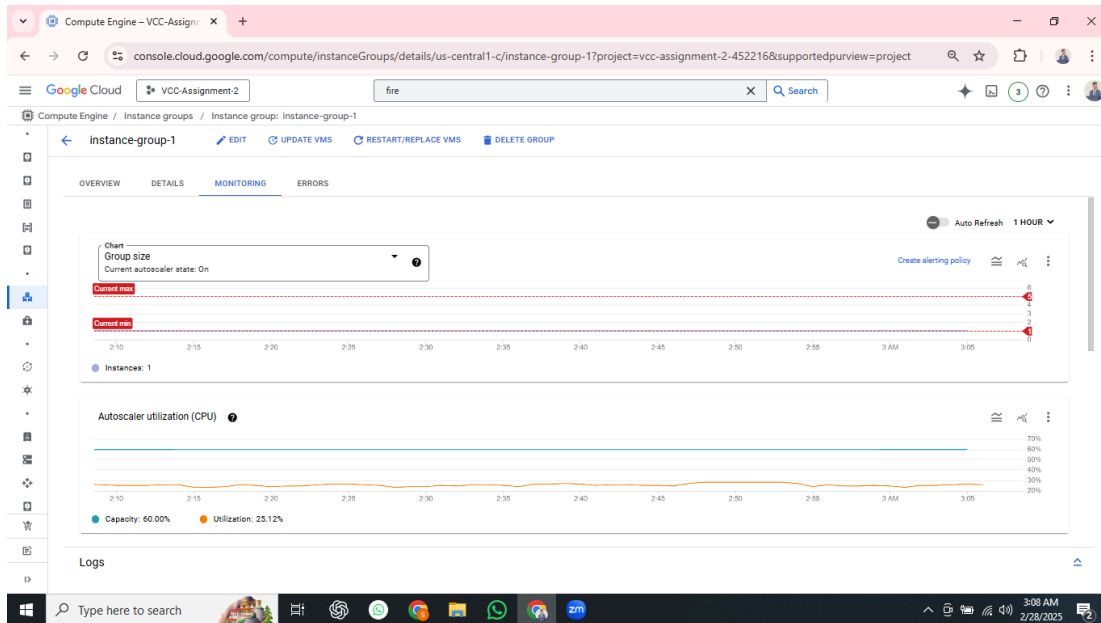
The screenshot shows the Google Cloud Console interface for an instance group named 'instance-group-1'. The page is titled 'Compute Engine - VCC-Assign' and the URL is 'console.cloud.google.com/compute/instanceGroups/details/us-central1-c/instance-group-1?project=vcc-assignment-2-452216&supportedpurview=project'. The left sidebar shows the navigation menu with 'Instance group: instance-group-1' selected. The main content area has tabs for 'OVERVIEW', 'DETAILS', 'MONITORING', and 'ERRORS'. The 'OVERVIEW' tab is active, showing the following information:

- Instances by status:** 1 instance (Ready)
- Instance by health:** Not configured (Autohealing off: [Configure](#))
- Autoscaling:** On (min 1, max 5) (Based on 1 metric and 0 schedules: [Configure](#))
- Status:** Ready
- Creation Time:** Feb 28, 2025, 12:45:27 AM UTC+05:30
- Description:**
- Target running size:** 1
- Template:** [instance-template-20250227-191034 \(Regional\)](#)
- Location:** us-central1-c
- Resize requests:** [NEW](#) None

Below the overview, there is a section for 'VM instances' with a table showing the details of the single instance in the group:

Status	Name	Creation Time	Template	Per instance config	Internal IP	External IP	Health Check Status	Connect
Ready	instance-group-1-dh1	Feb 28, 2025, 12:45:46 AM UTC+05:30	instance-template-20250227-191034 (Regional)		10.128.0.3 (nic0)	35.202.135.8		SSH

A notification banner at the bottom of the console says 'Stopping vcc-virtual-machine...'.



3.3 Implementing Security Measures

Security is critical when managing cloud resources. Implementing IAM roles and firewall rules ensures that access to resources is well-managed and protected.

3.3.1 Setting Up IAM Roles

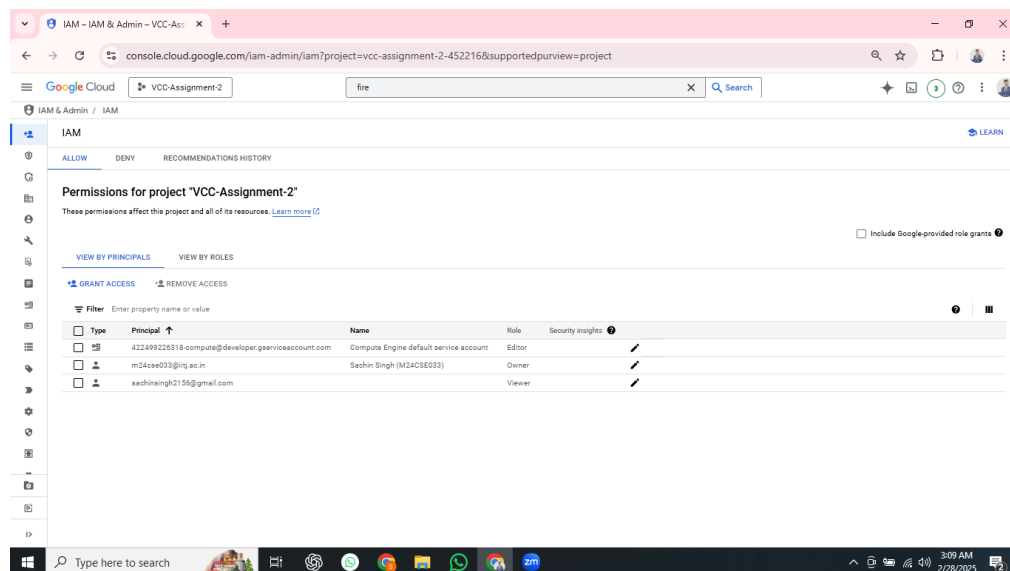
IAM (Identity and Access Management) allows role-based access control for better security.

Step 1: Navigate to IAM

1. Go to IAM & Admin > IAM.
2. Click + Add to add a new user.

Step 2: Assign Roles

1. **Enter Member Email.**
2. Assign roles:
 - **Compute Viewer:** Read-only access.
 - **Compute Admin:** Full control over Compute Engine resources.
3. Click **Save**.



3.3.2 Configuring Firewall Rules

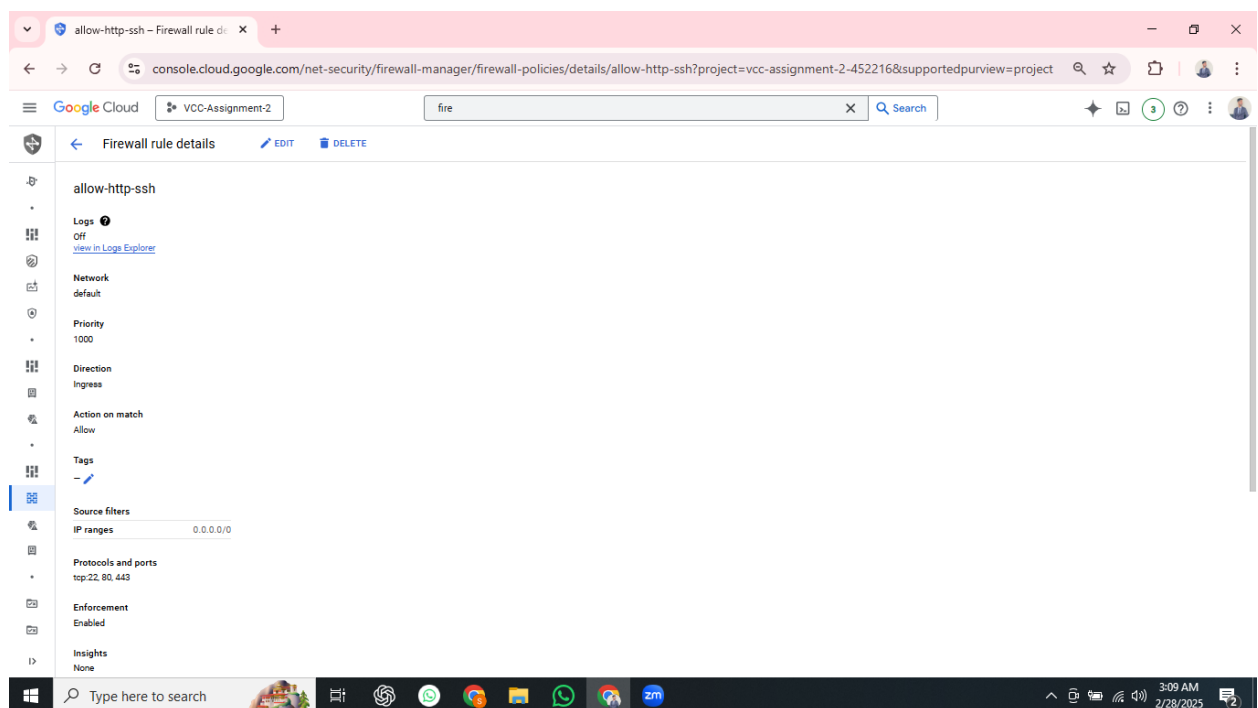
Firewall rules define which traffic is allowed to reach instances.

Step 1: Create a Firewall Rule

1. Navigate to VPC Network > Firewall.
2. Click Create Firewall Rule.

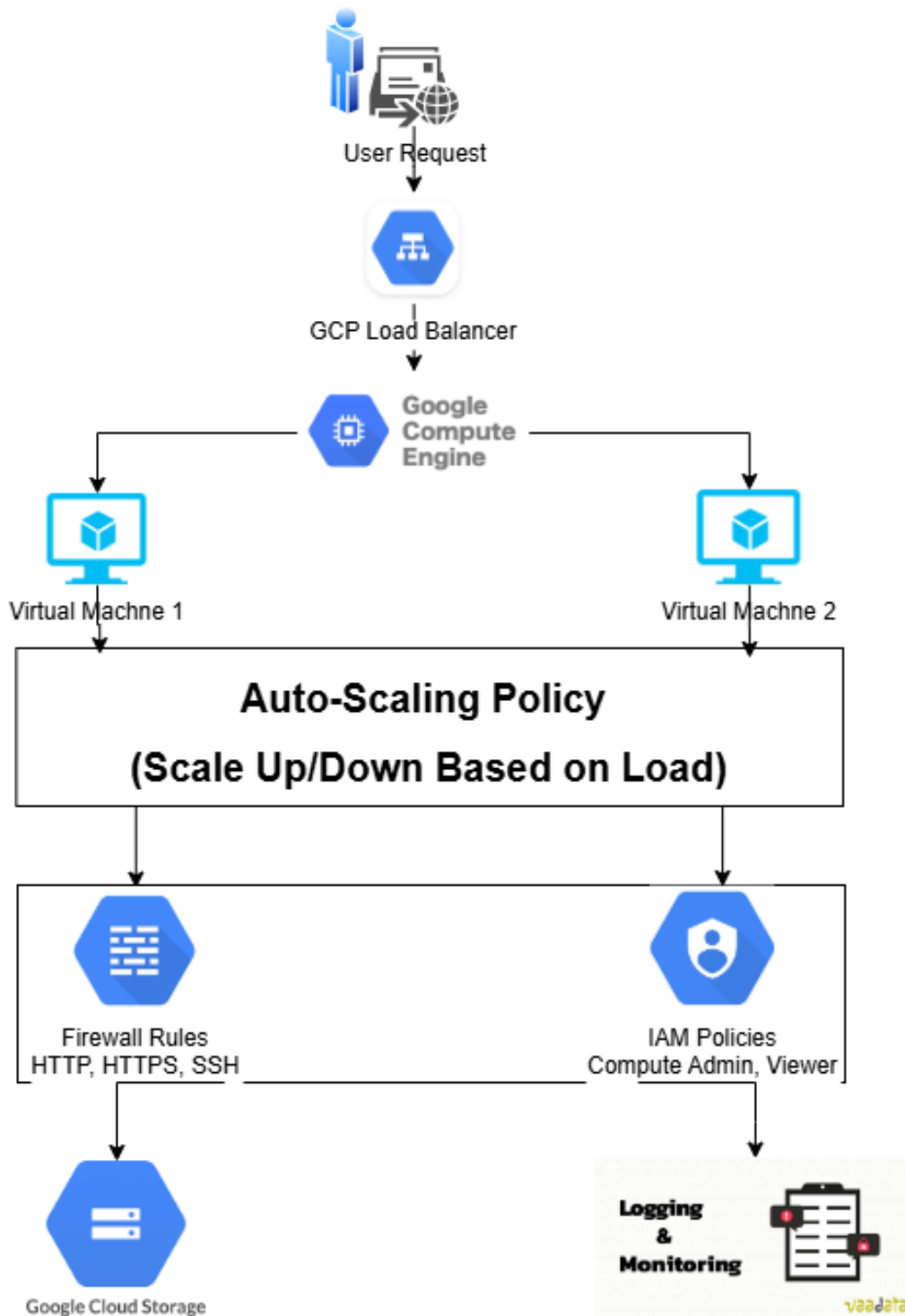
Step 2: Configure Firewall Rule

1. **Enter Rule Name:** allow-http-ssh
2. **Target:** Apply to all instances.
3. **Source IPv4 Range:** 0.0.0.0/0 (or restrict as needed).
4. **Allowed Protocols & Ports:**
 - **TCP:** 22, 80, 443
5. Click **Create**.



4. Architecture Design

The following architecture diagram visually represents the interaction between the VM, auto-scaling policies, firewall settings, and IAM roles.



Explanation of Architecture Diagram

1. **User Requests:** Incoming traffic from users is directed to the GCP Load Balancer.
2. **Load Balancer:** Distributes requests efficiently among multiple VM instances.
3. **Google Compute Engine (VMs):** The primary compute resources handling the user workload.
4. **Auto-Scaling Policy:** Automatically scales instances up/down based on CPU utilization.
5. **Firewall Rules:** Defines security rules for allowing HTTP, HTTPS, and SSH access.
6. **IAM Policies:** Ensures restricted access through Compute Admin and Viewer roles.
7. **Google Cloud Storage:** Used for storing logs and backups.
8. **Logging & Monitoring:** Helps in performance tracking and security monitoring.

5. Running a Python Program on the VM

Purpose of Implementing a Python Program

Verification (checking) is required to ensure that the virtual machine is correctly configured after the installation of both the virtual machine and the source code. Testing inside the VM is employed to check that the application is connected to the internet, the necessary packages have started, and can be run, and that the application is able to correctly perform a task relevant to real life. The shell script is a web crawler which extracts from the most visited web sites the headings of the papers. The resultant VM to be topologically complete and produce network requests correctly is guaranteed.

Implementation Steps

1. **Cloning the Repository Containing the Python Script:**

```
git clone https://github.com/your-repository-link.git
cd your-repository-folder
```

2. **Install Necessary Dependencies:**

```
sudo apt update
sudo apt install python3-pip -y
```


pip3 install requests beautifulsoup4

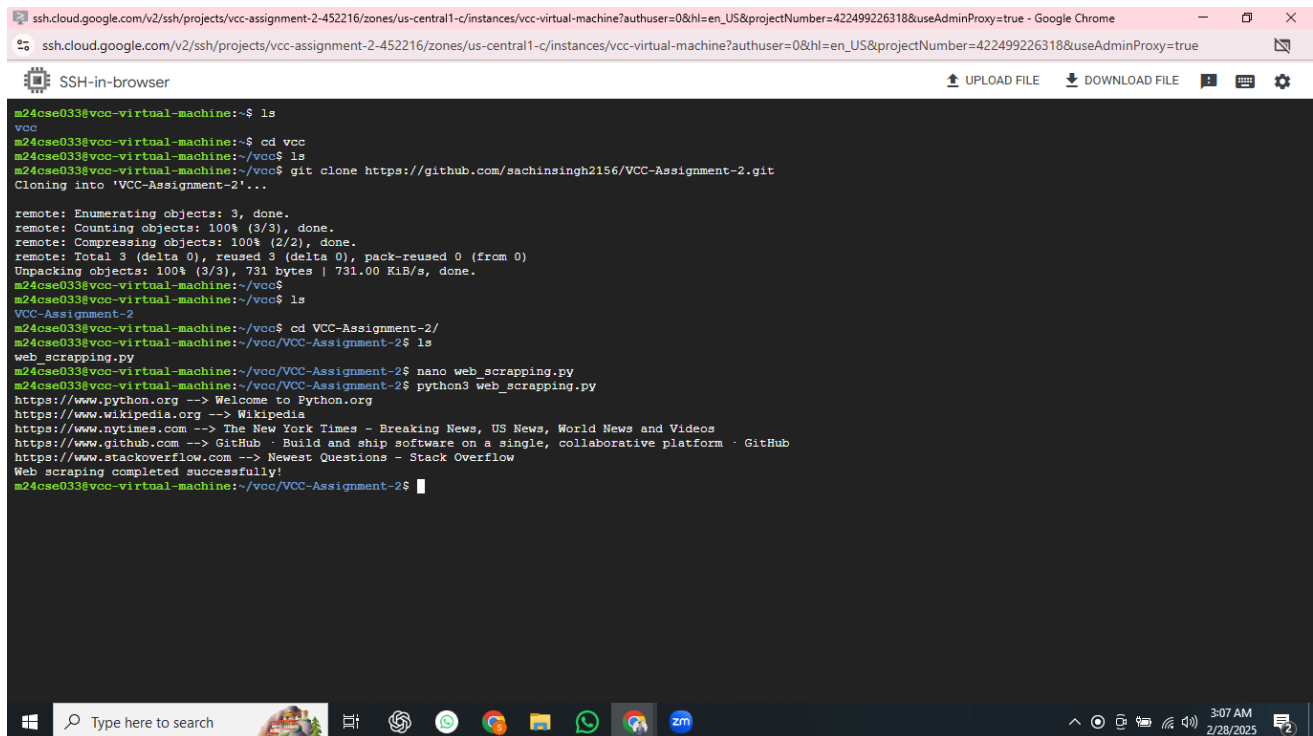
3. Run the Python Web Scraping Script:

```
python3 web_scraper.py
```

4. Expected Output:

```
https://www.wikipedia.org --> Wikipedia
https://www.python.org --> Welcome to Python.org
https://www.github.com --> GitHub · Build and ship software on a sing
https://www.nytimes.com --> The New York Times - Breaking News, US Ne
https://www.stackoverflow.com --> Newest Questions - Stack Overflow
Web scraping completed successfully!
```

This output confirms that the VM is successfully executing Python scripts, has internet access, and can perform automated tasks.



The screenshot shows an SSH-in-browser terminal window. The terminal output is as follows:

```
m24cse033@voc-virtual-machine:~$ ls
voc
m24cse033@voc-virtual-machine:~$ cd voc
m24cse033@voc-virtual-machine:~/voc$ ls
m24cse033@voc-virtual-machine:~/voc$ git clone https://github.com/sachinsingh2156/VCC-Assignment-2.git
Cloning into 'VCC-Assignment-2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 731 bytes | 731.00 KiB/s, done.
m24cse033@voc-virtual-machine:~/voc$ 
m24cse033@voc-virtual-machine:~/voc$ ls
VCC-Assignment-2
m24cse033@voc-virtual-machine:~/voc$ cd VCC-Assignment-2/
m24cse033@voc-virtual-machine:~/voc/VCC-Assignment-2$ ls
web_scrapping.py
m24cse033@voc-virtual-machine:~/voc/VCC-Assignment-2$ nano web_scrapping.py
m24cse033@voc-virtual-machine:~/voc/VCC-Assignment-2$ python3 web_scrapping.py
https://www.python.org --> Welcome to Python.org
https://www.wikipedia.org --> Wikipedia
https://www.nytimes.com --> The New York Times - Breaking News, US News, World News and Videos
https://www.github.com --> GitHub · Build and ship software on a single, collaborative platform · GitHub
https://www.stackoverflow.com --> Newest Questions - Stack Overflow
Web scraping completed successfully!
m24cse033@voc-virtual-machine:~/voc/VCC-Assignment-2$
```

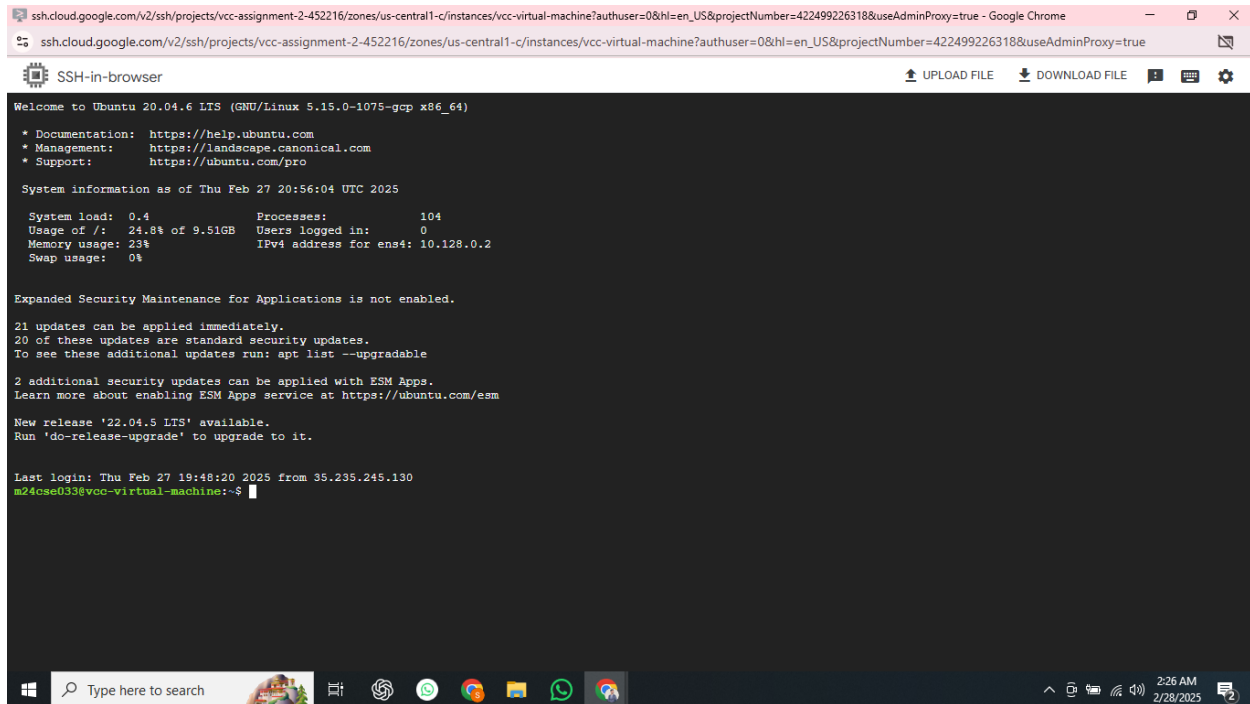
6. Source Code Repository

All deployment scripts and configurations are stored in the following GitHub repository:

<https://github.com/sachinsingh2156/VCC-Assignment-2.git>

7. Conclusion

This report offers highest-level, standardized, sequential guidance for the deployment of a Virtual Machine (VM) on GCP in addition to auto-scaling and security configuration. The security compliance, the resource management, the high availability and the cloud resources can be ensured as follows. The architecture is scalable to any scale, the performance is optimized, and security is ensured in the cloud environments.



The screenshot shows a terminal window titled "SSH-in-browser" with a dark background. The terminal output displays the Ubuntu 20.04.6 LTS welcome message, system information, and update notifications. The system information includes system load, processes, memory usage, and network details. Update notifications indicate that 21 updates can be applied immediately, with 20 being standard security updates. A new release '22.04.5 LTS' is also mentioned as available.

```
ssh.cloud.google.com/v2/ssh/projects/vcc-assignment-2-452216/zones/us-central1-c/instances/vcc-virtual-machine?authuser=0&hl=en_US&projectNumber=422499226318&useAdminProxy=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/vcc-assignment-2-452216/zones/us-central1-c/instances/vcc-virtual-machine?authuser=0&hl=en_US&projectNumber=422499226318&useAdminProxy=true

SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1075-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Feb 27 20:56:04 UTC 2025

System load:  0.4               Processes:    104
Usage of /:   24.8% of 9.51GB    Users logged in: 0
Memory usage: 23%              IPv4 address for ens4: 10.128.0.2
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.
20 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Feb 27 19:48:20 2025 from 35.235.245.130
m24cse033@vcc-virtual-machine:~$
```

References

1. **Google Cloud Documentation - Compute Engine**
<https://cloud.google.com/compute/docs>
2. **Google Cloud IAM & Security**
<https://cloud.google.com/iam/docs>
3. **Google Cloud Auto-Scaling**
<https://cloud.google.com/compute/docs/autoscaler>
4. **Firewall Rules in GCP**
<https://cloud.google.com/vpc/docs/firewalls>
5. **Python Web Scraping with BeautifulSoup**
<https://www.crummy.com/software/BeautifulSoup/>
6. **GitHub Repository Management**
<https://docs.github.com/en/get-started>

Appendix

1. **Video Presentation Link :**
<https://drive.google.com/file/d/1-oEEXT2lwhaFPseKNxAYPaxxtXM84POx/view?usp=sharing>
2. **PPT Link:**
<https://docs.google.com/presentation/d/1RzAQHK5lOoeLwOtN0Wc3uVYk5vILF28L/edit?usp=sharing&ouid=110909031381492795570&rtpof=true&sd=true>
3. **System Architecture Diagram:**
https://drive.google.com/file/d/1spQ4_FVAJUDE23ku-jr5a-wWVYZSx7_2/view?usp=sharing
4. **GitHub Repository Link:**
<https://github.com/sachinsingh2156/VCC-Assignment-2.git>