Virtualization and Cloud computing

Amresh Kumar M23CSA004

Assignment- 02

Github link:https://github.com/m23csa004/vcc_assignment2
Video link:[m23csa004_vcc_assign2.mkv]

Introduction

This report outlines the process of setting up a Virtual Machine (VM) on Google Cloud Platform (GCP), implementing auto-scaling policies based on CPU utilization, and configuring security measures such as IAM roles and firewall rules. The goal is to create a scalable and secure cloud infrastructure that adjusts dynamically to workload changes while ensuring controlled access and protection against unauthorized traffic.

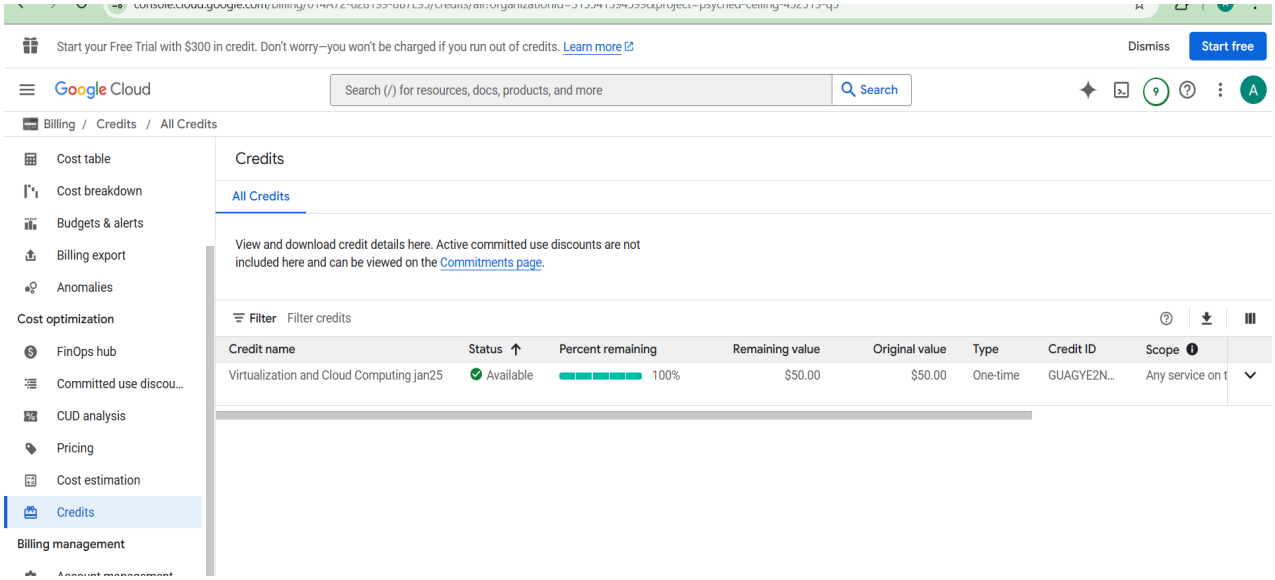
1. Getting GCP Credits using college account

Why Obtain GCP Credits?

GCP credits provide free access to cloud resources for a limited period, allowing users to experiment and deploy services without incurring immediate costs.

Step 1: Redeeming GCP Credits

1. Use the Cloud Platform Education Grants link provided by the Sir in the classroom.
2. Enter your name and email address on the page and click on submit .
3. A confirmation email will be sent for verification.
4. After verification, you receive a coupon code and a link to redeem it.
5. Upon successful redemption, you can see the credit balance(50$) in the Billing section of the GCP Console.



2. Creating a VM Instance on GCP

Why Use GCP Virtual Machines?

Google Compute Engine (GCE) provides scalable and flexible virtual machines that can run various applications. GCE offers benefits such as global infrastructure, automated backups, and cost optimization through sustained use discounts.

Step 1: Log in to Google Cloud Console

1. Navigate to Google Cloud Console.
2. Select your project or create a new one.
3. Select your project or create a new one. Alternatively, directly create a VM instance, and GCP will automatically assign the project as **My First Project.**

Step 2: Create a Virtual Machine

1. Go to Compute Engine > VM Instances.
2. Click Create Instance.
3. Configure the instance:
    - Name: Enter a name for the VM.
    - Region & Zone: Select the preferred region and zone.
    - Machine Type: Choose the desired CPU(I have chosen E2, because it uses less credit and I have no requirements of high computing power as of now) and memory configuration.
    - Boot Disk: Select an OS (e.g., Ubuntu 22.04 LTS).

    - Firewall Rules: Allow HTTP/HTTPS if needed.
4. Click Create to launch the VM.

3. Configuring Auto-Scaling Policies

Why Use Auto-Scaling?

Auto-scaling ensures that cloud resources dynamically scale based on demand. This helps optimize cost and performance by automatically adding or removing instances as CPU usage fluctuates.
It automatically creates the instance of vm if required and when the workload is decreased it closes all the created vm instances.

Step 1: Create a Managed Instance Group

1. Navigate to Compute Engine > Instance Groups.
2. Click Create Instance Group and select Managed Instance Group.
3. Configure the settings:
    - Base Instance Name: Define a name pattern or can select the existing vm.
    - Instance Template: Select an existing one or create a new one(I have created one and shown in video,).
    - Autoscaling Mode: Enable autoscaling.

Step 2: Set Auto-Scaling Policies

1. Configure scaling based on CPU utilization:
    - Minimum instances: Set a lower limit(1).
    - Maximum instances: Define an upper limit(i have set as 12).
    - Target CPU Utilization: Example: 60%(default, adjustable).
2. Click Create.

Observation of Auto-Scaling in Action

Running a CPU load test (as done in this case) increases utilization, triggering auto-scaling to create additional instances. In this scenario, four virtual instances were created as the CPU load increased.

4. Simulating Load on the VM

Why Simulate Load?

Load testing helps verify the effectiveness of auto-scaling policies and ensures that the infrastructure can handle high traffic scenarios without performance degradation.

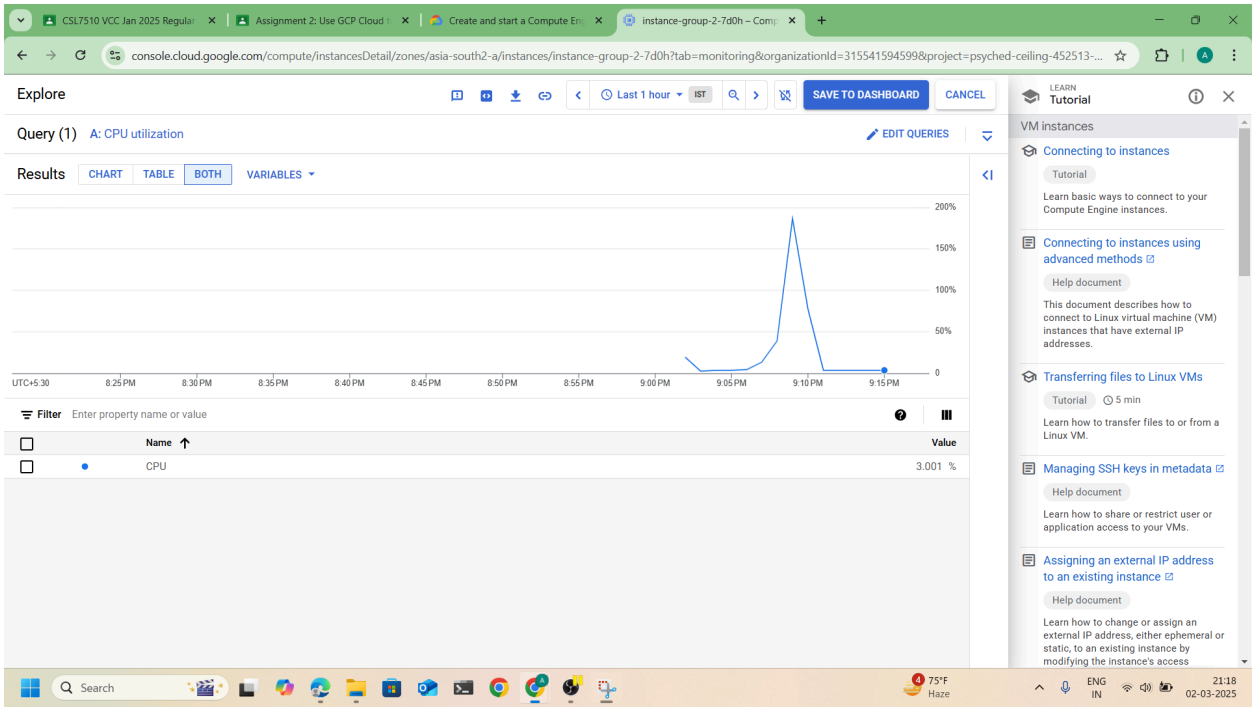
I have performed only a cpu load test, other load tests like memory, networ and disk can be performed.

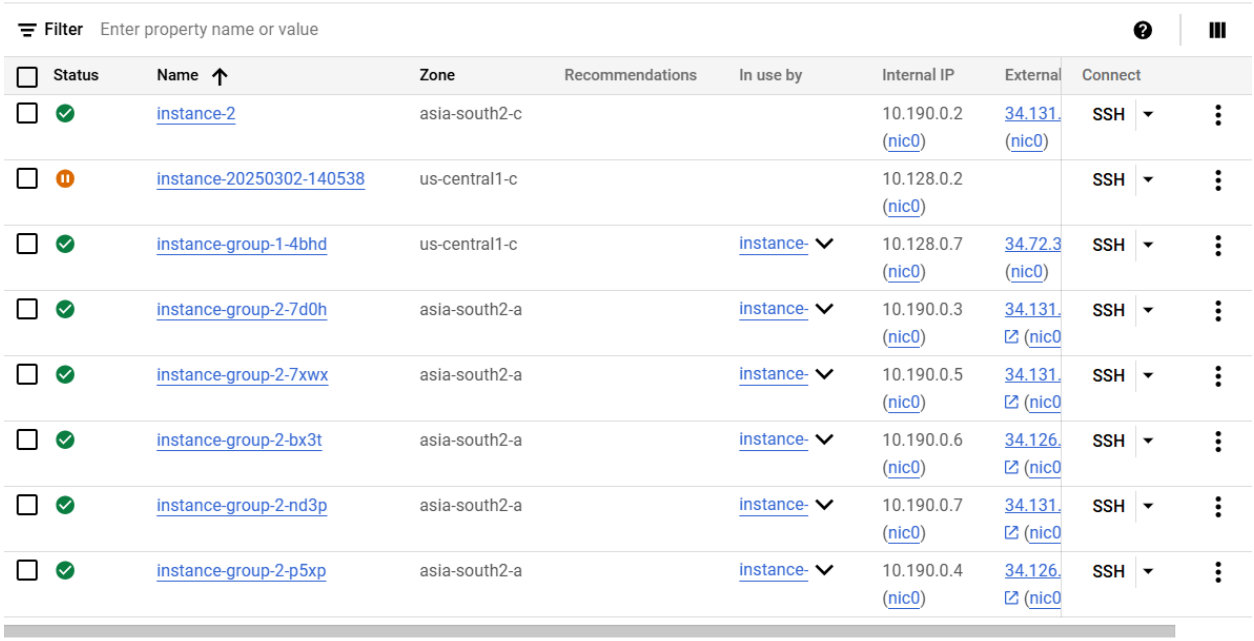
CPU Load Test commands

sudo apt update && sudo apt install -y stress

stress --cpu 4 --timeout 90

Here - - cpu 4 uses 4 cpu cores and
- - timeout 90 runs for 90 sec , I have used stress tool , you can also use infinite loops in python or any other operation that uses your cpu above 60% to show new instances of vm is being created.
Based on the load given through the **stress tool**, it created **5 instances of vm**.
The CPU monitoring and additional created vm instances screenshot is attached here.





Filter Enter property name or value

Status	Name	Zone	Recommendations	In use by	Internal IP	External	Connect
✓	instance-2	asia-south2-c			10.190.0.2 (nic0)	34.131. (nic0)	SSH
⏸	instance-20250302-140538	us-central1-c			10.128.0.2 (nic0)		SSH
✓	instance-group-1-4bhd	us-central1-c		instance-	10.128.0.7 (nic0)	34.72.3 (nic0)	SSH
✓	instance-group-2-7d0h	asia-south2-a		instance-	10.190.0.3 (nic0)	34.131. (nic0	SSH
✓	instance-group-2-7xwx	asia-south2-a		instance-	10.190.0.5 (nic0)	34.131. (nic0	SSH
✓	instance-group-2-bx3t	asia-south2-a		instance-	10.190.0.6 (nic0)	34.126. (nic0	SSH
✓	instance-group-2-nd3p	asia-south2-a		instance-	10.190.0.7 (nic0)	34.131. (nic0	SSH
✓	instance-group-2-p5xp	asia-south2-a		instance-	10.190.0.4 (nic0)	34.126. (nic0	SSH

5. Implementing Security Measures

Why Implement Security Measures?

Security configurations ensure that only authorized users access the cloud infrastructure while preventing unauthorized traffic and potential cyber threats.

Step 1: Setting Up IAM Roles for Restricted Access

1. Go to IAM & Admin > IAM.
2. Click Add to grant access.
3. Enter the user's email and assign roles:
    - Viewer: Read-only access.
    - Compute Admin: Full VM control.
    - Custom Role: Define custom permissions.
4. Click Save.

I have given access to m23csa019@iitj.ac.in as viewer role , so he can view my logs only.

Step 2: Configuring Firewall Rules

During VM instance creation, **HTTP and HTTPS traffic** were allowed. Additional firewall rules can be configured for advanced security.

1. Navigate to VPC Network > Firewall.
2. Click Create Firewall Rule and set the following:

    - Name: Enter a rule name.
    - Network: Choose the relevant VPC.
    - Direction: Ingress (incoming) or Egress (outgoing).
    - Targets: Select All instances in the network or specify instances.
    - Source IP Ranges: Define allowed IPs.
    - Protocols and Ports: Specify TCP, UDP, or ICMP rules.
3. Click Create.

Conclusion

This report provides a step-by-step guide for setting up a GCP VM, implementing auto-scaling policies, and applying essential security measures. Through auto-scaling and simulated load testing, the infrastructure dynamically adjusted to workload changes. Additionally, IAM roles and firewall rules ensured restricted access and enhanced security. These configurations help in optimizing performance while maintaining cost efficiency and security compliance.Architecture Diagram has been added in the next page.

[Scroll Below]

Architecture Diagram