VCC : Assignment No. 3

- By
Mandar Bhalerao (M23CSA025)

Google Drive Link :

https://drive.google.com/drive/folders/16SEBr1mXfLErhmXOmsAwGlzrT4PiUhbp?usp=sharing

Github Link :

git@github.com:m23csa025/VCC-Assignments.git
https://github.com/m23csa025/VCC-Assignments/tree/VCC-Asgnmt--03

Create a Local VM and Auto-Scale It to GCP or Any Other Public Cloud When Resource Usage Exceeds 75%

Objective:

Create a local VM and implement a mechanism to monitor resource usage. Configure it to auto-scale to a public cloud (e.g., GCP, AWS, or Azure) when resource usage exceeds 75%.

Deliverables:

1. Document Report:
    - Step-by-Step Instructions for Implementation:
    - Creation of a local VM (using VirtualBox, VMware, or similar).
    - Implementation of resource monitoring (using tools like Prometheus, Grafana, or a custom script).
    - Configuration of cloud auto-scaling policies and resource migration.
    - Deployment of a sample application to demonstrate the process.

2. Architecture Design:
    - Diagram illustrating the flow: local VM resource monitoring → scaling to cloud → deployment.

3. Link to Source Code Repo:
    - Repository containing monitoring scripts, deployment configurations, and application code.

4. Link to Recorded Video Demo: A video showing the setup, resource monitoring, and auto-scaling process in action, with a detailed voice-over.

1. Introduction

The process of creation of environment in Debian 12 based VM and autoscaling to GCP on CPU usage 75% are given below.

a. **Creation of Local VM**
   - My Laptop Macbook M2
   - Installation of UTM
   - Installation of Debian 12
   - Creation of New VM (vm3 - 4 GB, 40 GB)

b. **Implementation of resource monitoring tools**
   - Open terminal
   - **nano monitor.sh** (monitoring tool for monitoring CPU and Memory Usage**)**

Open monitor.sh ~/

app.py | appmigrate.sh | shift2gcp.sh | monitor.sh

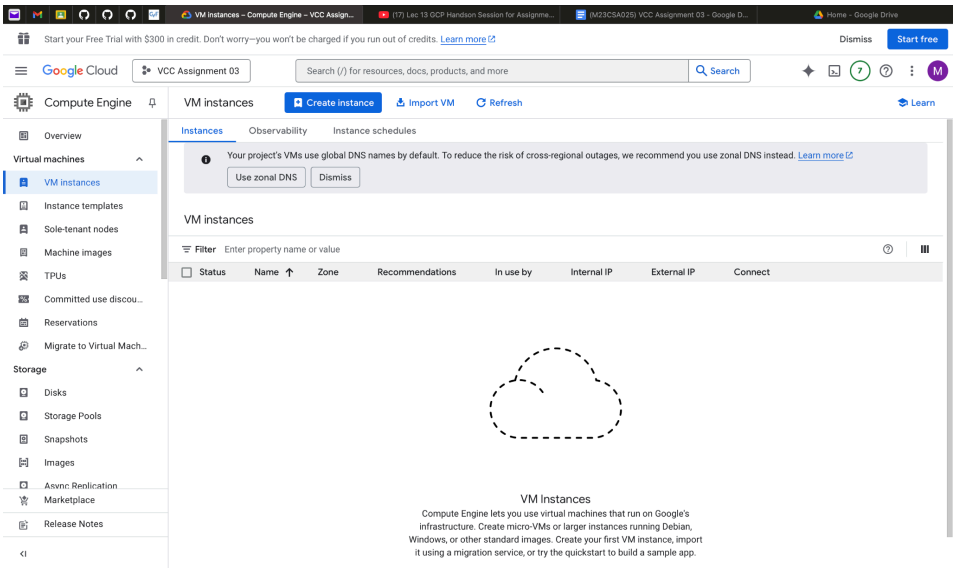
```
#!/bin/bash

#Monitor CPU and Memory Usage in Debian12

echo "CPU and Memory Usage Monitor (Debian 12)"
echo "Press[CTRL+C] to stop"

while true;do
    # Get CPU usage using 'top'
    CPU_USAGE=$(top -bn1|grep "Cpu(s)" | awk '{print $2+$4}')

    #Get Memory usage using 'free'

    MEM_TOTAL=$(free -m| awk 'NR==2{print $2}')
    MEM_USED=$(free -m | awk 'NR==2{print $3}')
    MEM_USAGE=$(echo "scale=2; $MEM_USED*100/$MEM_TOTAL"| bc)

    #Display the results

    echo "CPU Usage: $CPU_USAGE%| Memory Usage: $MEM_USAGE%"

    #Sleep for 1 second before repeating
    sleep 1
done
```

- **Install stress app to load cpu**

  - stress –cpu 4 –timeout 60s

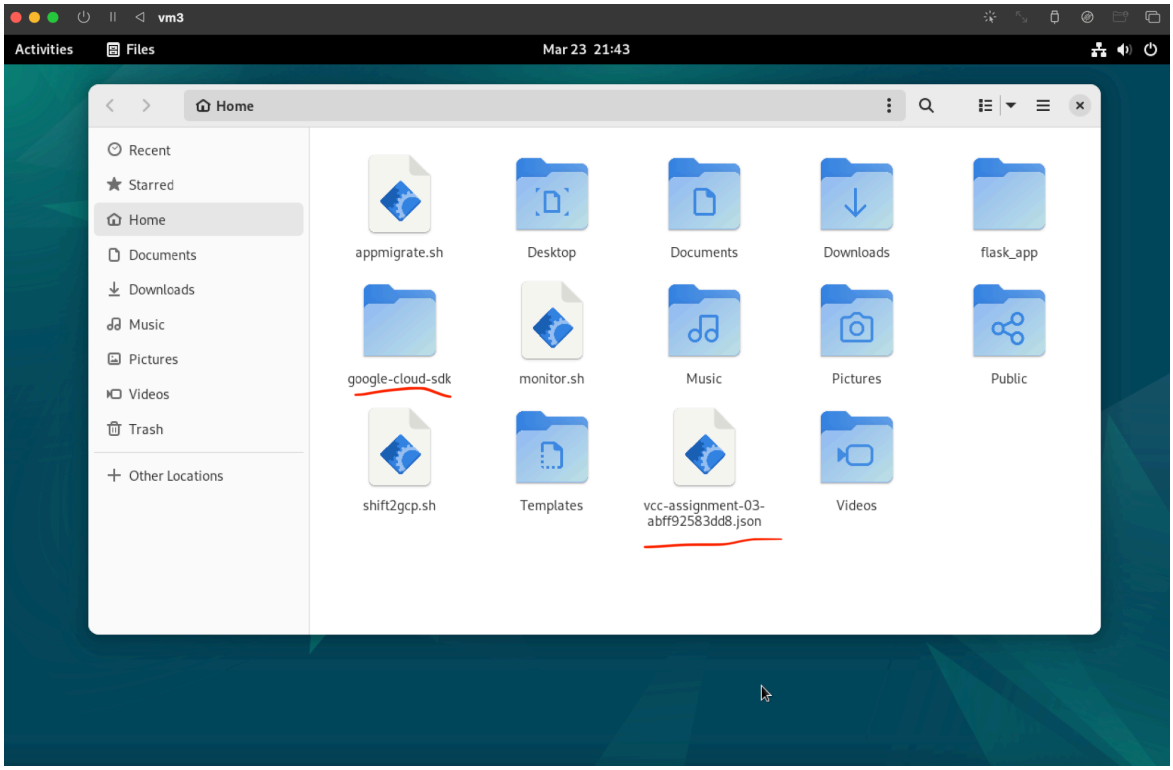
c. Configuration of GCP migration policies & resource migration

- **Current status of VMs in the GCP accts**



- **nano shift2gcp.sh** (code for autoscaling to GCP on cpu load reaching 75% threshold)

- **Activation of service acct on GCP and Creation of Key**



- **Installation of Google SDK (as shown above)**

d. Deployment of sample application and demonstration

- creation of application **(app.py)**

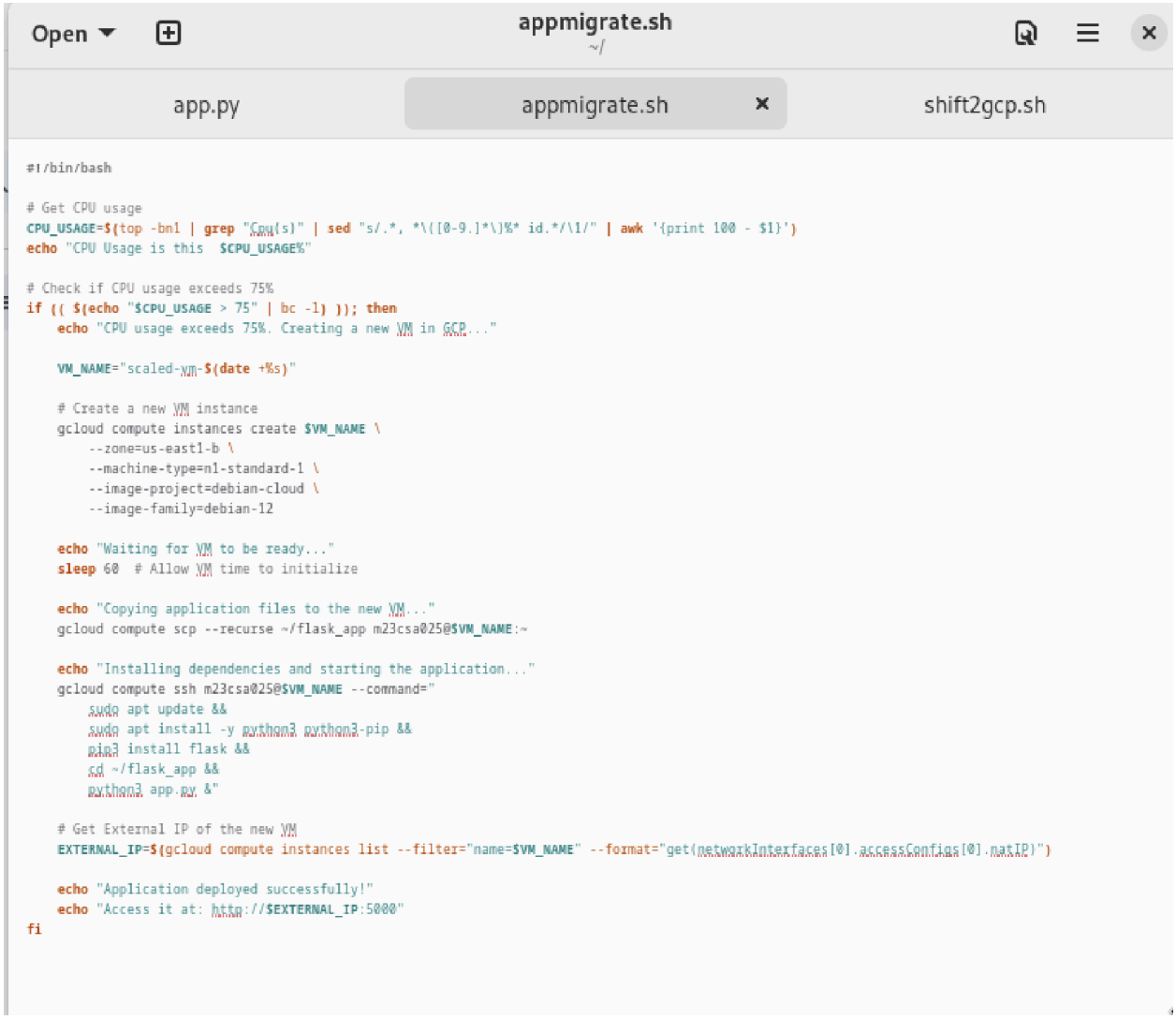
Open

app.py
~/flask_app

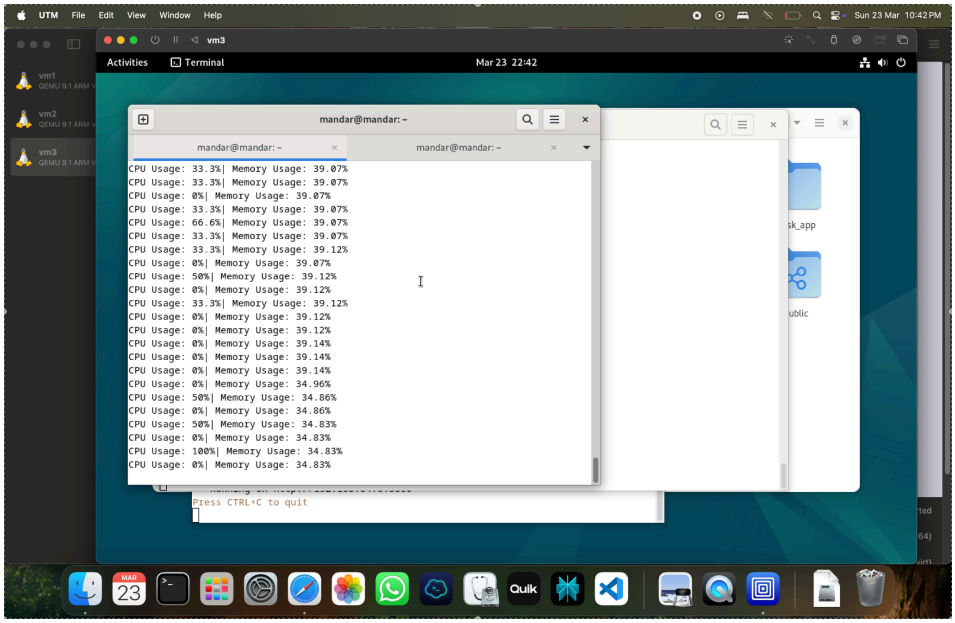
app.py | appmigrate.sh | shift2gcp.sh

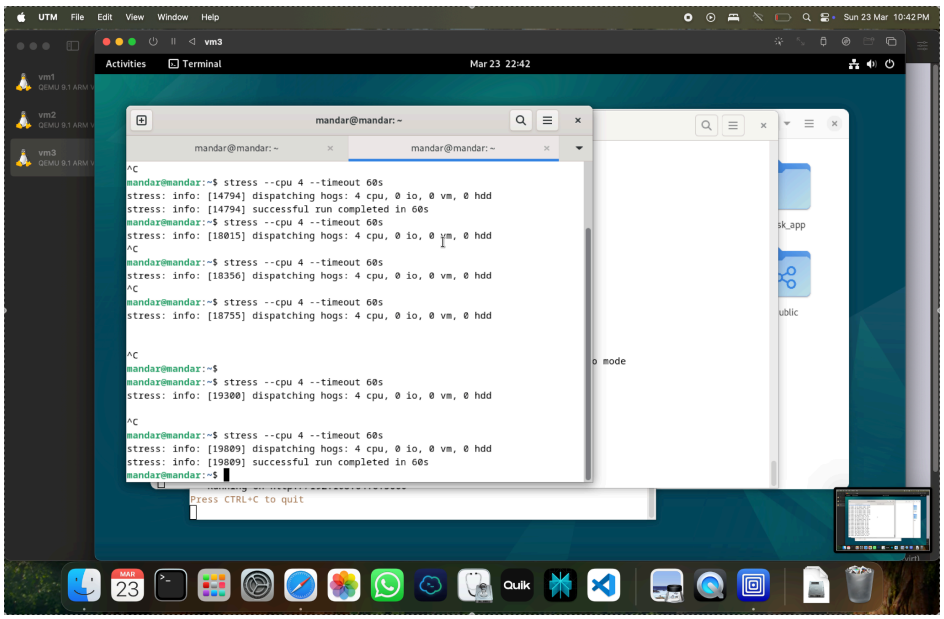
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hi.. This is Mandar (M23CSA025) from IITJ"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

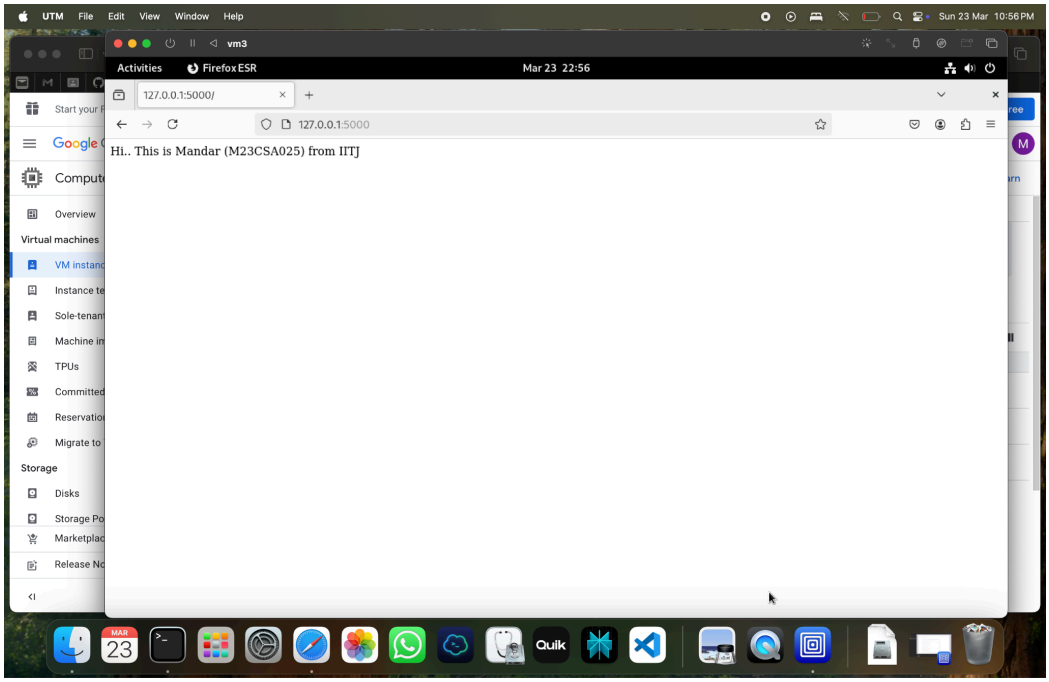
- Loading of Python, flask on vm3
- Creation of application migration code (**appmigrate.sh)**

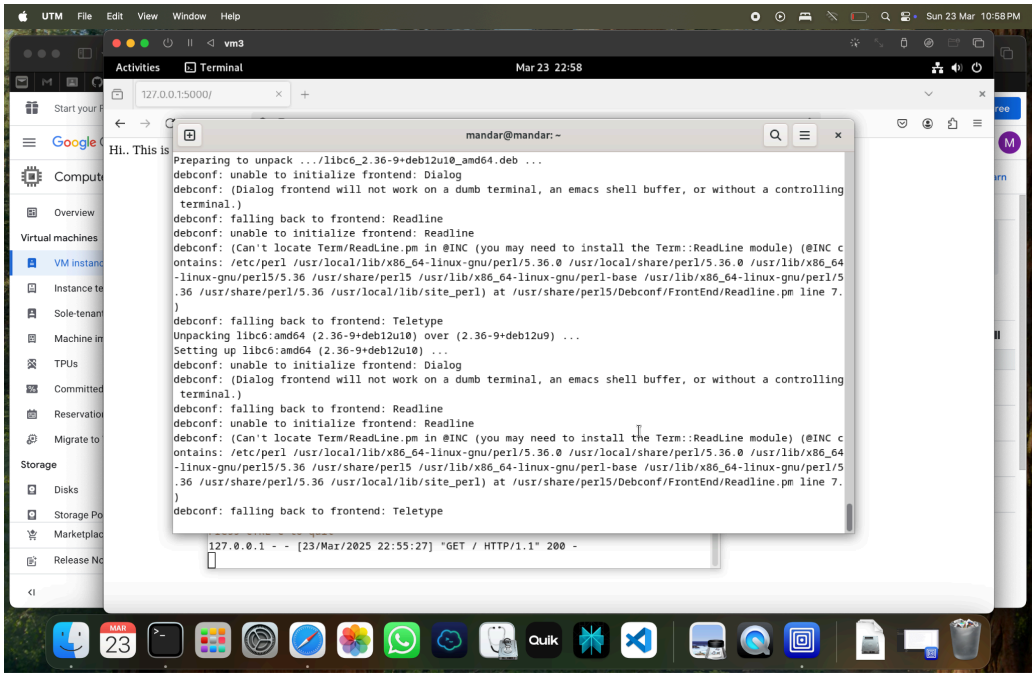

Open
appmigrate.sh
~/
app.py
appmigrate.sh
shift2gcp.sh
#!/bin/bash
# Get CPU usage
CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | sed "s/.*, *\([0-9.]*\)%* id.*/\1/" | awk '{print 100 - $1}')
echo "CPU Usage is this $CPU_USAGE%"
# Check if CPU usage exceeds 75%
if (($(echo "$CPU_USAGE > 75" | bc -l))); then
echo "CPU usage exceeds 75%. Creating a new VM in GCP..."
VM_NAME="scaled-vm-$(date +%s)"
# Create a new VM instance
gcloud compute instances create $VM_NAME \
--zone=us-east1-b \
--machine-type=n1-standard-1 \
--image-project=debian-cloud \
--image-family=debian-12
echo "Waiting for VM to be ready..."
sleep 60 # Allow VM time to initialize
echo "Copying application files to the new VM..."
gcloud compute scp --recurse ~/flask_app m23csa025@$VM_NAME:~
echo "Installing dependencies and starting the application..."
gcloud compute ssh m23csa025@$VM_NAME --command="
sudo apt update &&
sudo apt install -y python3 python3-pip &&
pip3 install flask &&
cd ~/flask_app &&
python3 app.py &"
# Get External IP of the new VM
EXTERNAL_IP=$(gcloud compute instances list --filter="name=$VM_NAME" --format="get(networkInterfaces[0].accessConfigs[0].natIP)")
echo "Application deployed successfully!"
echo "Access it at: http://$EXTERNAL_IP:5000"
fi

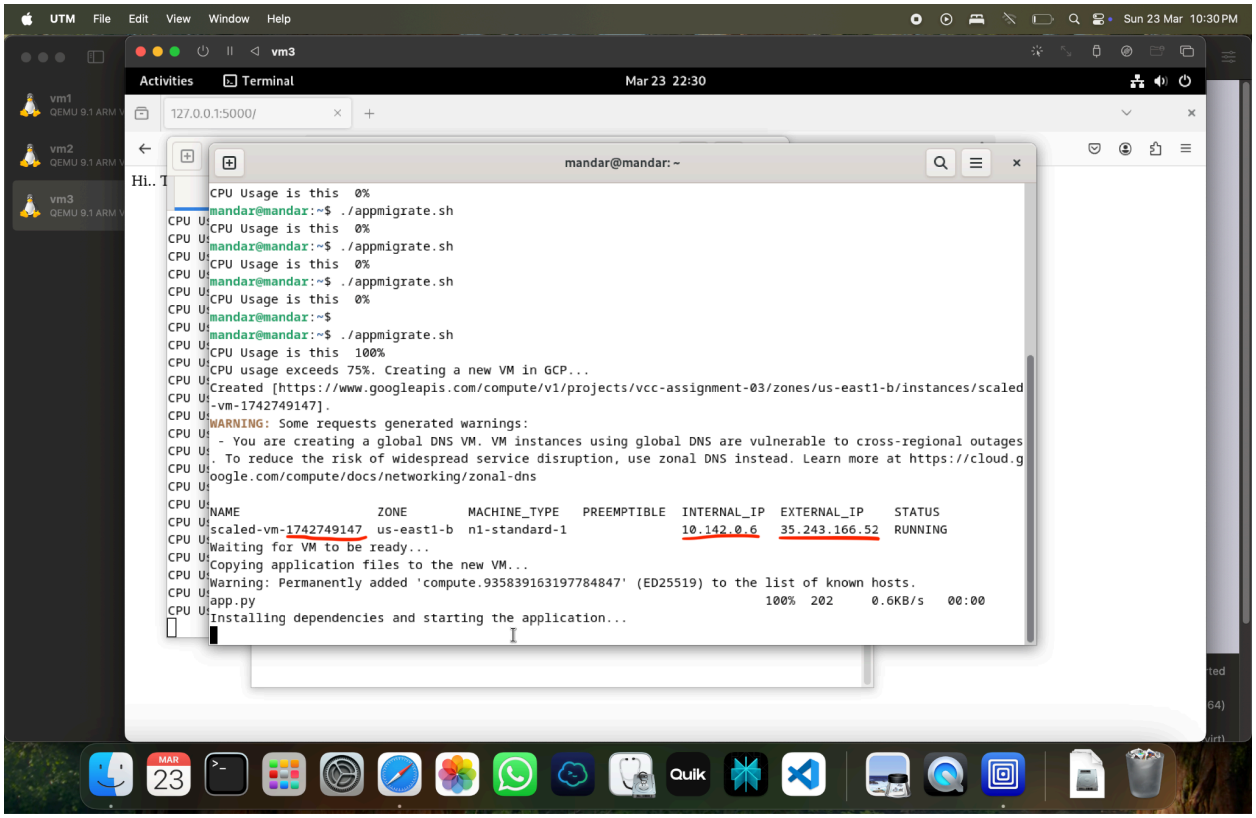
Execution

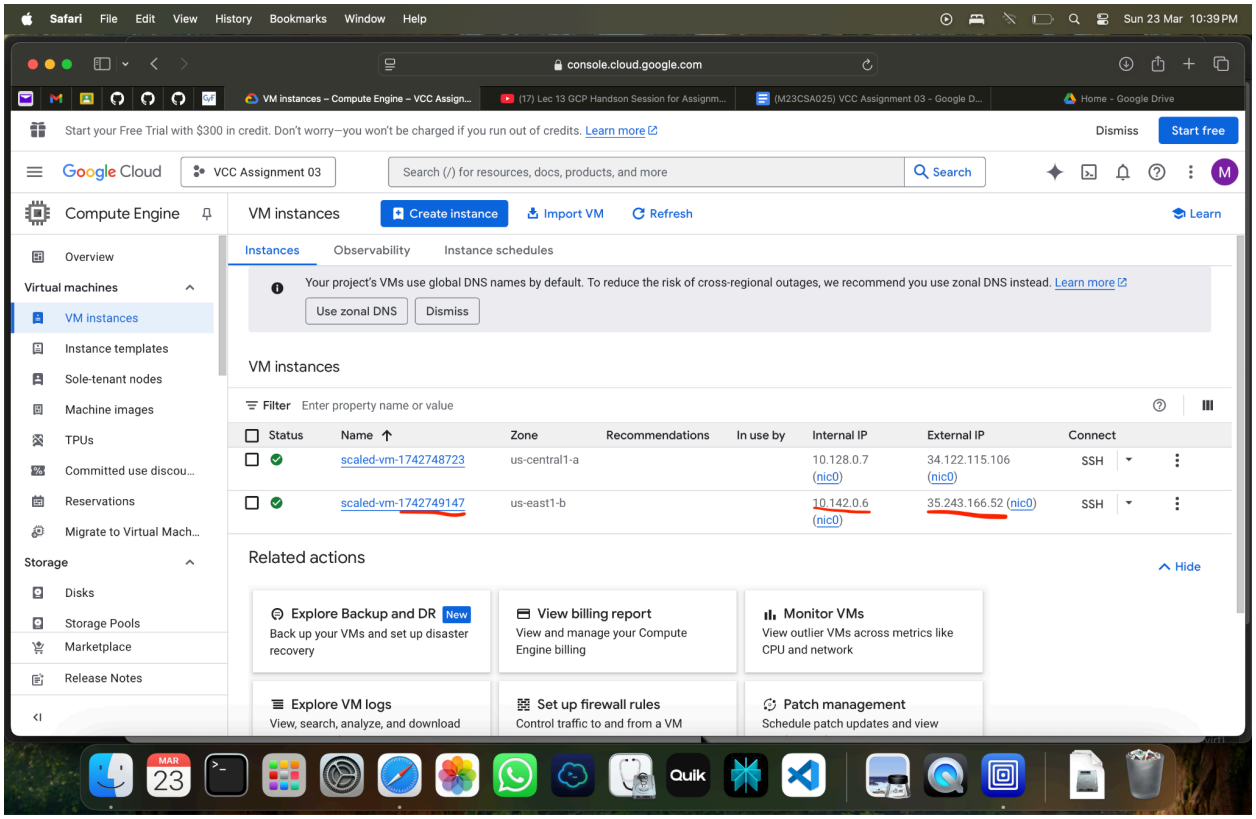

UTM File Edit View Window Help
Sun 23 Mar 10:42 PM
vm3
Activities
Terminal
Mar 23 22:42
mandar@mandar: ~
mandar@mandar: ~
mandar@mandar: ~
CPU Usage: 33.3%| Memory Usage: 39.07%
CPU Usage: 33.3%| Memory Usage: 39.07%
CPU Usage: 0%| Memory Usage: 39.07%
CPU Usage: 33.3%| Memory Usage: 39.07%
CPU Usage: 66.6%| Memory Usage: 39.07%
CPU Usage: 33.3%| Memory Usage: 39.07%
CPU Usage: 33.3%| Memory Usage: 39.12%
CPU Usage: 0%| Memory Usage: 39.07%
CPU Usage: 50%| Memory Usage: 39.12%
CPU Usage: 0%| Memory Usage: 39.12%
CPU Usage: 33.3%| Memory Usage: 39.12%
CPU Usage: 0%| Memory Usage: 39.12%
CPU Usage: 0%| Memory Usage: 39.12%
CPU Usage: 0%| Memory Usage: 39.14%
CPU Usage: 0%| Memory Usage: 39.14%
CPU Usage: 0%| Memory Usage: 39.14%
CPU Usage: 0%| Memory Usage: 34.96%
CPU Usage: 50%| Memory Usage: 34.86%
CPU Usage: 0%| Memory Usage: 34.86%
CPU Usage: 50%| Memory Usage: 34.83%
CPU Usage: 0%| Memory Usage: 34.83%
CPU Usage: 100%| Memory Usage: 34.83%
CPU Usage: 0%| Memory Usage: 34.83%
Press CTRL+C to quit


UTM File Edit View Window Help
Sun 23 Mar 10:42 PM
vm3
Activities
Terminal
Mar 23 22:42
mandar@mandar: ~
mandar@mandar: ~
mandar@mandar: ~
^C
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [14794] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress: info: [14794] successful run completed in 60s
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [18015] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
^C
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [18356] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
^C
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [18755] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
^C
mandar@mandar:~$
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [19300] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
^C
mandar@mandar:~$ stress --cpu 4 --timeout 60s
stress: info: [19809] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress: info: [19809] successful run completed in 60s
mandar@mandar:~$
Press CTRL+C to quit


UTM File Edit View Window Help
Sun 23 Mar 10:56 PM
vm3
Activities
Firefox ESR
Mar 23 22:56
127.0.0.1:5000/
127.0.0.1:5000
Hi.. This is Mandar (M23CSA025) from IITJ


UTM File Edit View Window Help
Sun 23 Mar 10:58 PM
vm3
Activities
Terminal
Mar 23 22:58
127.0.0.1:5000/
mandar@mandar: ~
Preparing to unpack .../libc6_2.36-9+deb12u10_amd64.deb ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.36.0 /usr/local/share/perl/5.36.0 /usr/lib/x86_64-linux-gnu/perl5/5.36 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl-base /usr/lib/x86_64-linux-gnu/perl/5.36 /usr/share/perl/5.36 /usr/local/lib/site_perl) at /usr/share/perl5/Debconf/FrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
Unpacking libc6:amd64 (2.36-9+deb12u10) over (2.36-9+deb12u9) ...
Setting up libc6:amd64 (2.36-9+deb12u10) ...
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.36.0 /usr/local/share/perl/5.36.0 /usr/lib/x86_64-linux-gnu/perl5/5.36 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl-base /usr/lib/x86_64-linux-gnu/perl/5.36 /usr/share/perl/5.36 /usr/local/lib/site_perl) at /usr/share/perl5/Debconf/FrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
127.0.0.1 - - [23/Mar/2025 22:55:27] "GET / HTTP/1.1" 200 -


UTM File Edit View Window Help
Sun 23 Mar 10:30 PM
vm3
Activities
Terminal
Mar 23 22:30
vm1
QEMU 9.1 ARM
vm2
QEMU 9.1 ARM
vm3
QEMU 9.1 ARM
127.0.0.1:5000/
mandar@mandar: ~
CPU Usage is this 0%
mandar@mandar:~$ ./appmigrate.sh
CPU Usage is this 0%
mandar@mandar:~$ ./appmigrate.sh
CPU Usage is this 0%
mandar@mandar:~$ ./appmigrate.sh
CPU Usage is this 0%
mandar@mandar:~$
mandar@mandar:~$ ./appmigrate.sh
CPU Usage is this 100%
CPU usage exceeds 75%. Creating a new VM in GCP...
Created [https://www.googleapis.com/compute/v1/projects/vcc-assignment-03/zones/us-east1-b/instances/scaled-vm-1742749147].
WARNING: Some requests generated warnings:
- You are creating a global DNS VM. VM instances using global DNS are vulnerable to cross-regional outages. To reduce the risk of widespread service disruption, use zonal DNS instead. Learn more at https://cloud.google.com/compute/docs/networking/zonal-dns
NAME ZONE MACHINE_TYPE PREEMPTIBLE INTERNAL_IP EXTERNAL_IP STATUS
scaled-vm-1742749147 us-east1-b n1-standard-1 10.142.0.6 35.243.166.52 RUNNING
Waiting for VM to be ready...
Copying application files to the new VM...
Warning: Permanently added 'compute.935839163197784847' (ED25519) to the list of known hosts.
app.py 100% 202 0.6KB/s 00:00
Installing dependencies and starting the application...


Safari File Edit View History Bookmarks Window Help
Sun 23 Mar 10:39 PM
console.cloud.google.com
Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more
Dismiss
Start free
Google Cloud
VCC Assignment 03
Search (/) for resources, docs, products, and more
Search
Compute Engine
VM instances
Create instance
Import VM
Refresh
Learn
Overview
Virtual machines
VM instances
Instance templates
Sole-tenant nodes
Machine images
TPUs
Committed use discou...
Reservations
Migrate to Virtual Mach...
Storage
Disks
Storage Pools
Marketplace
Release Notes
Instances
Observability
Instance schedules
Your project's VMs use global DNS names by default. To reduce the risk of cross-regional outages, we recommend you use zonal DNS instead. Learn more
Use zonal DNS
Dismiss
VM instances
Filter Enter property name or value
Status Name Zone Recommendations In use by Internal IP External IP Connect
scaled-vm-1742748723 us-central1-a 10.128.0.7 (nic0) 34.122.115.106 (nic0) SSH
scaled-vm-1742749147 us-east1-b 10.142.0.6 (nic0) 35.243.166.52 (nic0) SSH
Related actions
Hide
Explore Backup and DR New
Back up your VMs and set up disaster recovery
View billing report
View and manage your Compute Engine billing
Monitor VMs
View outlier VMs across metrics like CPU and network
Explore VM logs
View, search, analyze, and download
Set up firewall rules
Control traffic to and from a VM
Patch management
Schedule patch updates and view

Architecture Design