Name: Naga Lakshmi Gurrala

SJSU ID:018236127 Assignment 1: Virtual Machine / VMM Setup

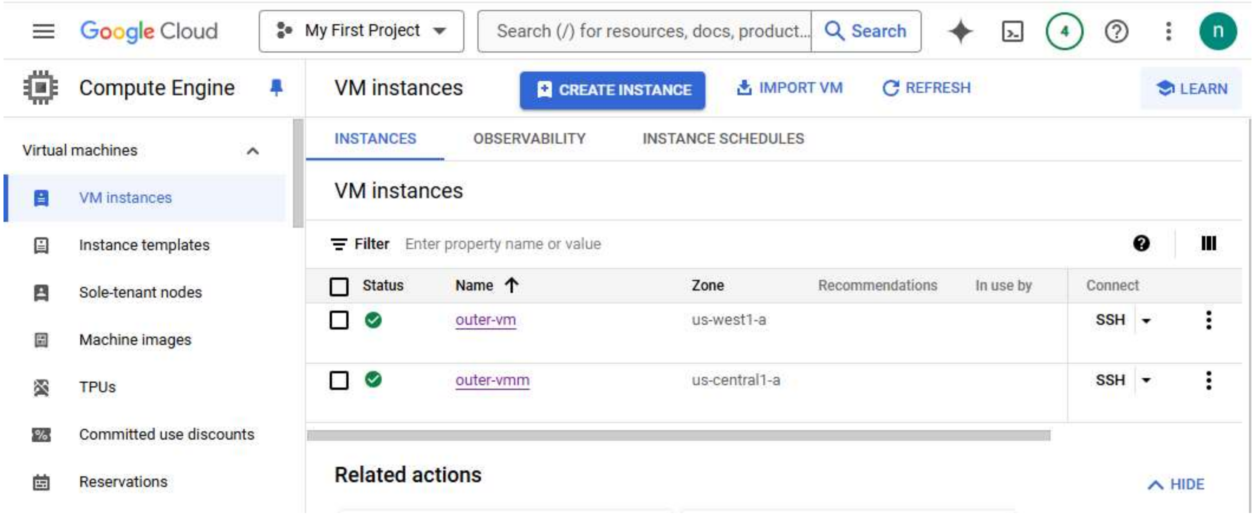
I have created a Virtual machine using GCP (Google cloud Platform). Using GCP simplifies handling nested virtualization requirements without overloading local hardware

1. Set Up Google Cloud for Nested Virtualization

- **Create a GCP Project**: If you haven't already, set up a new project in the Google Cloud Console.
- **Enable Billing**: Ensure billing is active to use virtual machines.
- **Enable the Compute Engine API**: Navigate to `APIs & Services > Library` and enable Compute Engine API for the project.

2. Configure and Launch the "Outer" Virtual Machine

- **Select a Machine Type**: Choose a machine type that supports nested virtualization (e.g., `n2-standard-2` or higher).
- **Enable Nested Virtualization**:
  - Go to `Compute Engine > VM instances`, click **Create Instance**.
  - Under **Machine configuration**, select a supported machine series, such as `N2` or `C2`.
  - In the **CPU platform and GPU** section, ensure **"Enable Nested Virtualization"** is selected.
- **Choose an OS Image**: Select Ubuntu as your OS (recommended for ease of use and community support).
- **Network and Firewall Settings**: Ensure SSH is enabled and other necessary ports are open for development.
- **Create the Instance**: Click **Create** to launch the outer VM.



Status	Name	Zone	Recommendations	In use by	Connect
✓	outer-vm	us-west1-a			SSH
✓	outer-vmm	us-central1-a			SSH

3. Install KVM on the Outer VM

- **SSH into the Outer VM**:
  - Use SSH from the GCP Console to connect to your outer VM.
- **Install KVM and Virtualization Tools**:

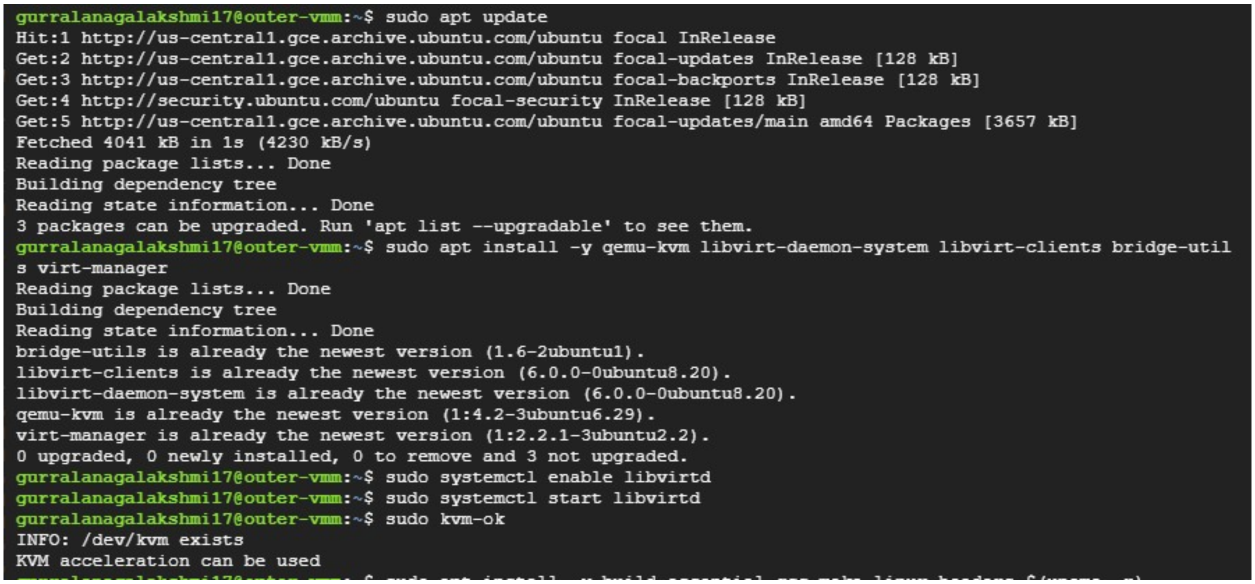
```
bash

sudo apt update
sudo apt install -y qemu-kvm libvirt-daemon-system libvirt-clients
bridge-utils virt-manager
```

- **Enable and Start KVM Services**:

```
bash

sudo systemctl enable libvirtd
sudo systemctl start libvirtd
```

- **Verify KVM Installation**:

```
bash

sudo kvm-ok
```

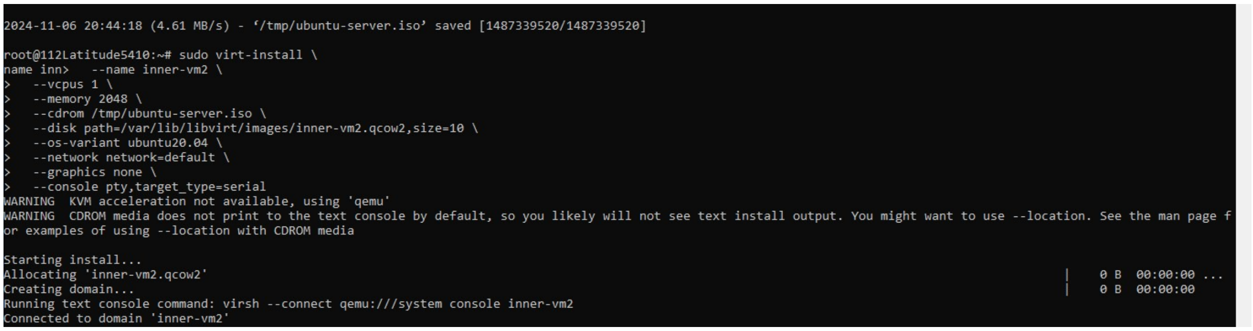


```
gurralanagalakshmi17@outer-vmm:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3657 kB]
Fetched 4041 kB in 1s (4230 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
gurralanagalakshmi17@outer-vmm:~$ sudo apt install -y qemu-kvm libvirt-daemon-system libvirt-clients bridge-util
s virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
bridge-utils is already the newest version (1.6-2ubuntu1).
libvirt-clients is already the newest version (6.0.0-0ubuntu8.20).
libvirt-daemon-system is already the newest version (6.0.0-0ubuntu8.20).
qemu-kvm is already the newest version (1:4.2-3ubuntu6.29).
virt-manager is already the newest version (1:2.2.1-3ubuntu2.2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
gurralanagalakshmi17@outer-vmm:~$ sudo systemctl enable libvirtd
gurralanagalakshmi17@outer-vmm:~$ sudo systemctl start libvirtd
gurralanagalakshmi17@outer-vmm:~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```

4. Create the “Inner” Virtual Machine within the Outer VM

- **Open Virt-Manager**:
  - Launch `virt-manager` on the outer VM to create the inner VM.
- **Set Up the Inner VM**:
  - Configure the inner VM as a typical virtual machine with Ubuntu or a similar Linux distribution.

- Allocate enough resources for the inner VM to run smoothly (e.g., 1 CPU, 1-2 GB RAM

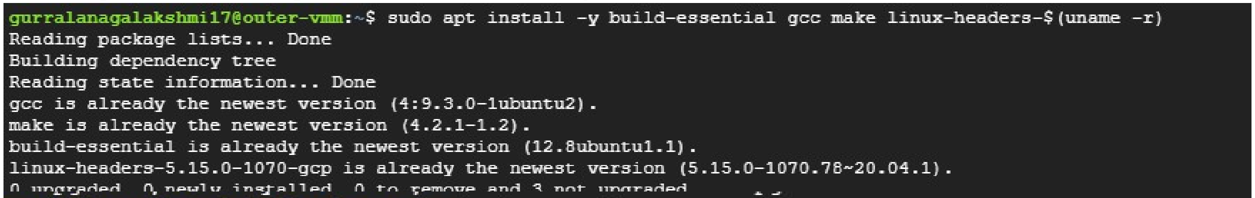


```
2024-11-06 20:44:18 (4.61 MB/s) - ‘/tmp/ubuntu-server.iso’ saved [1487339520/1487339520]

root@112Latitude5410:~# sudo virt-install \
name inn>   --name inner-vm2 \
>   --vcpus 1 \
>   --memory 2048 \
>   --cdrom /tmp/ubuntu-server.iso \
>   --disk path=/var/lib/libvirt/images/inner-vm2.qcow2,size=10 \
>   --os-variant ubuntu20.04 \
>   --network network=default \
>   --graphics none \
>   --console pty,target_type=serial
WARNING  KVM acceleration not available, using 'qemu'
WARNING  CDROM media does not print to the text console by default, so you likely will not see text install output. You might want to use --location. See the man page f
or examples of using --location with CDROM media

Starting install...
Allocating 'inner-vm2.qcow2'                                                                                                                        |    0 B  00:00:00 ...
Creating domain...                                                                                                                                  |    0 B  00:00:00
Running text console command: virsh --connect qemu:///system console inner-vm2
Connected to domain 'inner-vm2'
```

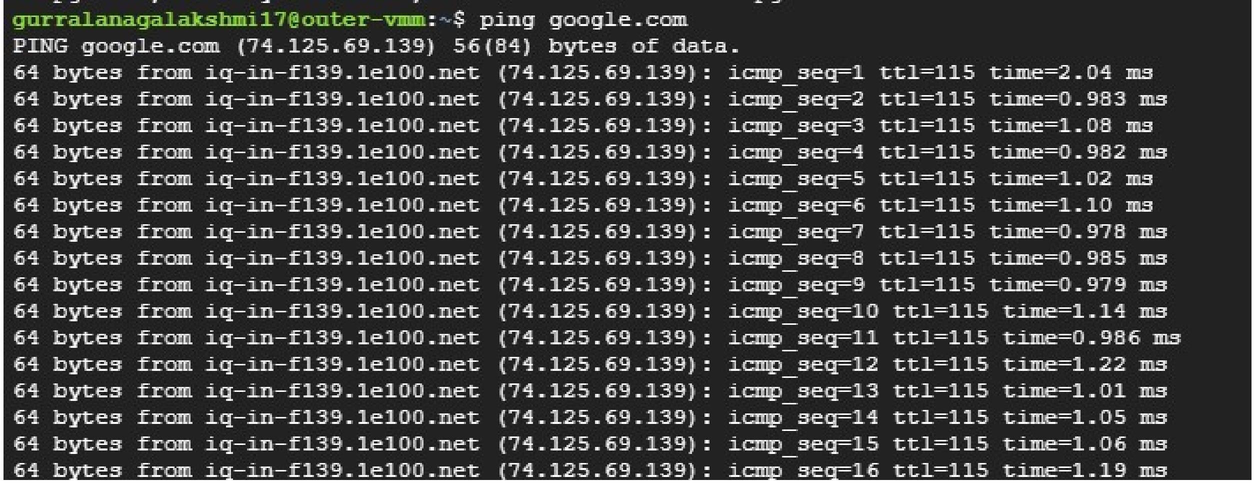
5. Install Development Tools on the Outer VM

- Install necessary tools for upcoming assignments:

```
bash

sudo apt install -y build-essential gcc make linux-headers-$(uname -r)
```



```
gurralanagalakshmi17@outer-vmm:~$ sudo apt install -y build-essential gcc make linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:9.3.0-1ubuntu2).
make is already the newest version (4.2.1-1.2).
build-essential is already the newest version (12.8ubuntu1.1).
linux-headers-5.15.0-1070-gcp is already the newest version (5.15.0-1070.78~20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```



```
gurralanagalakshmi17@outer-vmm:~$ ping google.com
PING google.com (74.125.69.139) 56(84) bytes of data.
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=1 ttl=115 time=2.04 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=2 ttl=115 time=0.983 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=3 ttl=115 time=1.08 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=4 ttl=115 time=0.982 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=5 ttl=115 time=1.02 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=6 ttl=115 time=1.10 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=7 ttl=115 time=0.978 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=8 ttl=115 time=0.985 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=9 ttl=115 time=0.979 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=10 ttl=115 time=1.14 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=11 ttl=115 time=0.986 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=12 ttl=115 time=1.22 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=13 ttl=115 time=1.01 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=14 ttl=115 time=1.05 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=15 ttl=115 time=1.06 ms
64 bytes from iq-in-f139.1e100.net (74.125.69.139): icmp_seq=16 ttl=115 time=1.19 ms
```