

Assignment 1: Virtual Machine / VMM Setup

Team Members : Junie Mariam Varghese and Rinku Tekchandani

We used Google Cloud Platform (GCP) to set up nested virtualization, as it allowed both of us to access the virtual machine (VM). Rinku researched the initial requirements to set up the outer VM and how to enable virtualization on it. Junie supported by configuring the subnet to ensure the network interfaces were correctly set up. We referred to the GCP documentation to enable nested virtualization on the VM, found here: [Enabling Nested Virtualization](#). Additionally, we followed this guide to create the nested VM: [Creating Nested VMs](#).

Each step required further research to find the correct commands to execute, as some commands in the standard GCP documentation didn't work as expected. Rinku extensively investigated why QEMU was used and suggested the correct parameters to start the inner VM. We also encountered an issue when trying to find and download a QEMU-compatible OS image for the L2 VM, as well as locating the login details. Rinku suggested an innovative solution to bypass the password and reset it, allowing us to log in to the inner VM successfully. Junie also handled other configurations on the outer VM, ensuring KVM was installed and running properly.

We faced a challenge when setting up external network access for the inner VM. Junie identified the need to revise the QEMU command and suggested manually setting the IP address and bypassing the network configuration to the inner VM to resolve the connectivity issues. Both of us explored various approaches to address this, testing different solutions to ensure proper network access. Through collaboration and persistence, we were able to implement the suggestion, and together we successfully overcame the issue. Ultimately, we documented each step and resolved the network configuration challenge.

1. Create a subnet in default networks

```
gcloud compute networks subnets create subnet1 \
  --network=default \
  --region=us-west1 \
  --range=10.0.0.0/24
```

2. Using the Subnet in Your L1 VM Creation:

```
gcloud compute instances create outer-vm
--enable-nested-virtualization \
  --tags http-server,https-server --can-ip-forward \
  --min-cpu-platform "Intel Haswell" \
  --network-interface subnet=subnet1,aliases=/30
```

3. Connect to the L1 VM (OuterVM)

```
gcloud compute ssh outer-vm
```

4. Update the VM and install the necessary packages:

```
sudo apt update && sudo apt install uml-utilities qemu-kvm  
bridge-utils virtinst libvirt-daemon-system libvirt-clients -y
```

5. Start the default network that comes with the **libvirt package:**

```
sudo virsh net-start default
```

6. Run the following command to check that you have the **virbr0 bridge:**

```
ip addr
```

```
rohan@router-vm:~$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 42:01:0a:00:00:02 brd ff:ff:ff:ff:ff:ff  
    altnam enp0s4  
    inet 10.0.0.2/32 brd 10.0.0.2 scope global dynamic ens4  
        valid_lft 3429sec preferred_lft 3429sec  
    inet 192.168.122.101/24 scope global ens4  
        valid_lft forever preferred_lft forever  
    inet6 fe80::4001:aff:fe00:2/64 scope link  
        valid_lft forever preferred_lft forever  
3: virbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000  
    link/ether 52:54:00:e5:7e:ab brd ff:ff:ff:ff:ff:ff  
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0  
        valid_lft forever preferred_lft forever  
4: tap0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master virbr0 state UP group default qlen 1000  
    link/ether 8a:7e:c6:d6:44:bd brd ff:ff:ff:ff:ff:ff  
    inet 192.168.122.101/24 scope global tap0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::887e:c6ff:fed6:44bd/64 scope link  
        valid_lft forever preferred_lft forever
```

7. Create a **tap interface to go from the L1 VM to the L2 VM:**

```
sudo tuncctl -t tap0  
sudo ifconfig tap0 up  
sudo brctl addif virbr0 tap0  
sudo brctl show  
  
sudo ip addr add 192.168.122.101/24 dev tap0  
sudo ip link set tap0 up
```

8. Download a QEMU-Compatible OS Image for the L2 VM

```
wget  
https://cloud-images.ubuntu.com/minimal/releases/focal/release/  
ubuntu-20.04-minimal-cloudimg-amd64.img -O l2-image.img
```

9. To preset the password:

```
sudo apt update
sudo apt install libguestfs-tools -y
```

```
sudo virt-customize -a l2-image.img --root-password password:root123
```

```
rohan@outer-vm:~$ sudo virt-customize -a l2-image.img --root-password password:root123
[ 0.0] Examining the guest ...
[ 22.3] Setting a random seed
virt-customize: warning: random seed could not be set for this type of
guest
[ 22.3] Setting the machine ID in /etc/machine-id
[ 22.4] Setting passwords
[ 23.8] Finishing off
```

10. Add the User to the kvm Group

```
sudo usermod -aG kvm $USER
And to verify it
ls -alh /dev/kvm
```

```
rohan@outer-vm:~$ sudo usermod -aG kvm $USER
rohan@outer-vm:~$ ls -alh /dev/kvm
crw-rw---- 1 root kvm 10, 232 Nov  6 20:32 /dev/kvm
```

11. Start the nested VM with this command

```
sudo qemu-system-x86_64 -enable-kvm -hda l2-image.img -m 512
-netdev tap,id=mynet0,ifname=tap0,script=no -device
virtio-net-pci,netdev=mynet0 -nographic
```

12. Log in into you nested VM L2:

```
ip addr
sudo ip addr add 192.168.122.10/24 dev ens3
sudo dhclient ens3
ip addr
ip route
```

```

root@ubuntu:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.10/24 scope global ens3
        valid_lft forever preferred_lft forever
    inet 192.168.122.76/24 brd 192.168.122.255 scope global secondary dynamic ens3
        valid_lft 2974sec preferred_lft 2974sec
    inet6 fe80::5054:ff:fe12:3456/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu:~#

```

13. Enabled external network access to L2 VM:

```

root@ubuntu:~# ping google.com
PING google.com (74.125.197.101) 56(84) bytes of data:
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=1 ttl=114 time=0.940 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=2 ttl=114 time=0.471 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=3 ttl=114 time=0.462 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=4 ttl=114 time=0.471 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=5 ttl=114 time=0.498 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=6 ttl=114 time=0.504 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=7 ttl=114 time=0.478 ms
64 bytes from pk-in-f101.1e100.net (74.125.197.101): icmp_seq=8 ttl=114 time=0.488 ms

```

14. Verify the L2 VM (InnerVM) status on the L1 VM(OuterVM)

```
ps aux | grep qemu
```

```

rohan@outer-vm:~$ ps aux | grep qemu
root      26130  0.0  0.1  9920  4472 pts/0    S+   22:43   0:00 sudo qemu-system-x86_64 -enable-kvm -hda 12-image.img -m 512 -netdev tap,id=mynt0,lframe=tap0,script=no -device virtio-net-pci,netdev=mynt0 -nographic
root      26131  4.1 15.5 1122180 584976 pts/0    Sl+  22:43   2:43 qemu-system-x86_64 -enable-kvm -hda 12-image.img -m 512 -netdev tap,id=mynt0,lframe=tap0,script=no -device virtio-net-pci,netdev=mynt0 -nographic
rohan     26543  0.0  0.0   5132   648 pts/2     S+   23:49   0:00 grep qemu

```