Name: Aldwin Joseph B. Revilla	**Date Performed: 08/22/2023**
Course/Section: BSCpE	**Date Submitted: 08/23/2023**
Instructor: Engr. Roman Richard	**Semester and SY: 1st sem / 2023-2024**

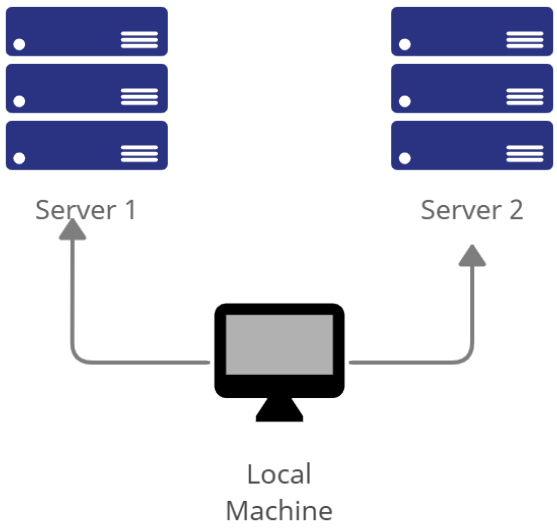
Activity 1: Configure Network using Virtual Machines

1. **Objectives:**
   1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox
   1.2. Set-up a Virtual Network and Test Connectivity of VMs

2. **Discussion:**

Network Topology:

Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).

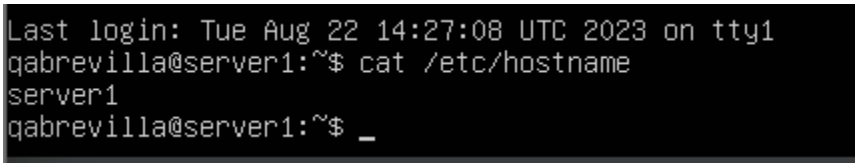


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1

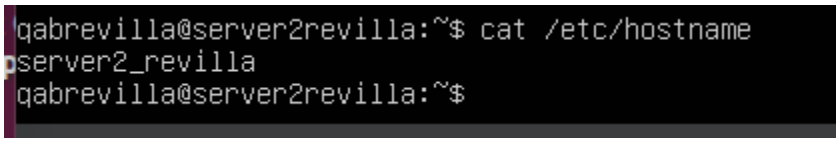
```
gabrevilla@server1revilla:~$ cat /etc/hostname
server1_revilla
```

before

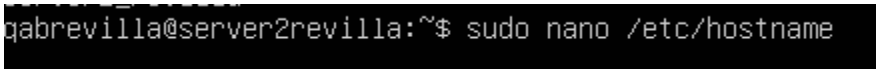


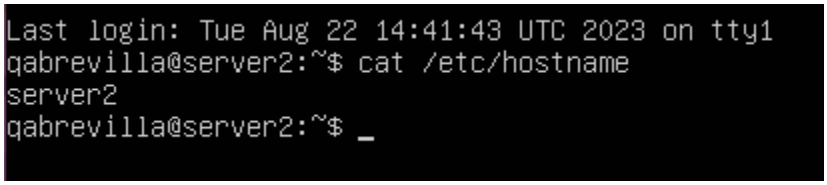
after

1.2 Use server2 for Server 2



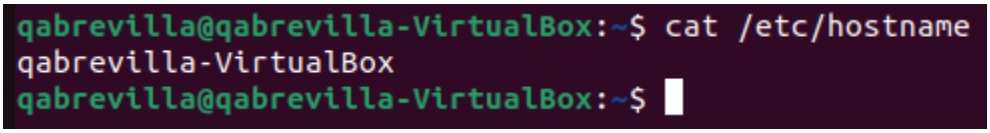
before



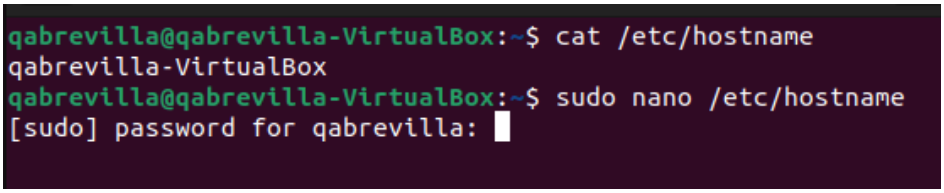


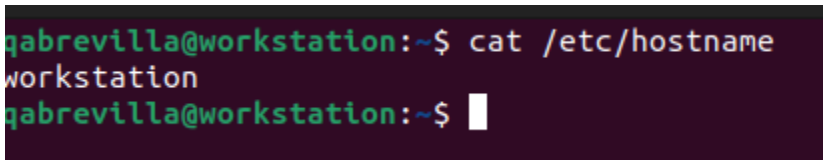
after

1.3 Use workstation for the Local Machine



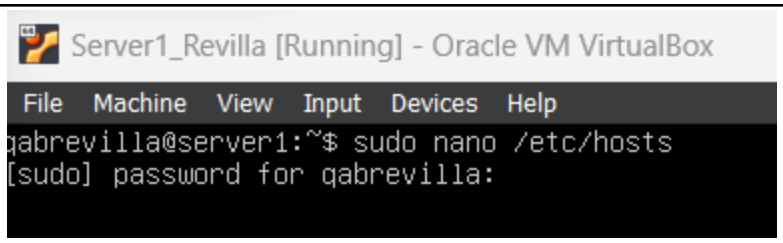
before

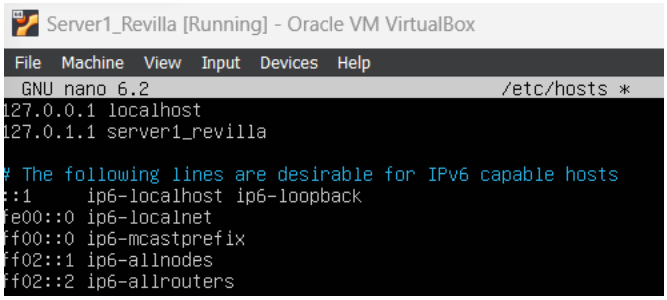




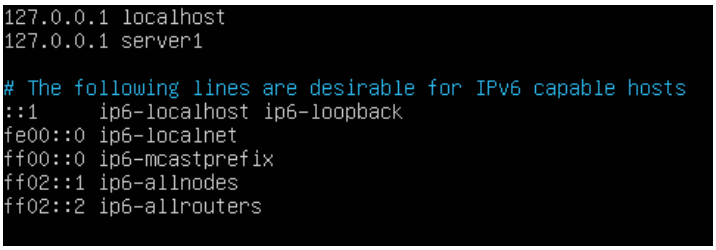
after

2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1



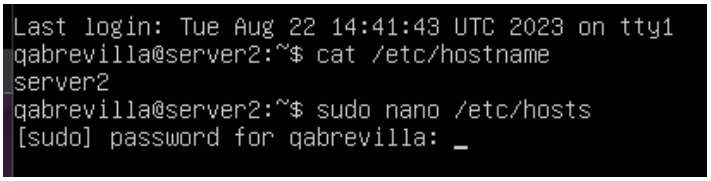


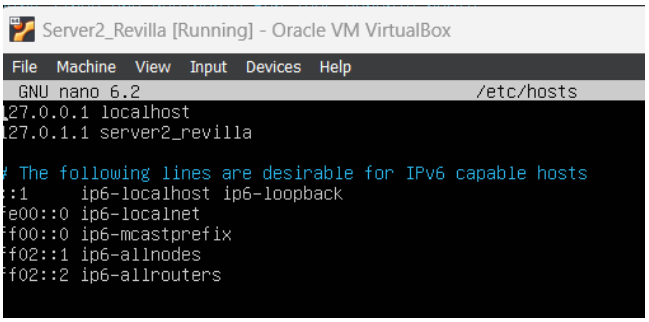
before



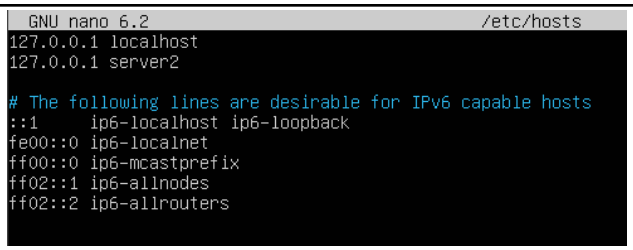
after

2.2 Type 127.0.0.1 server 2 for Server 2



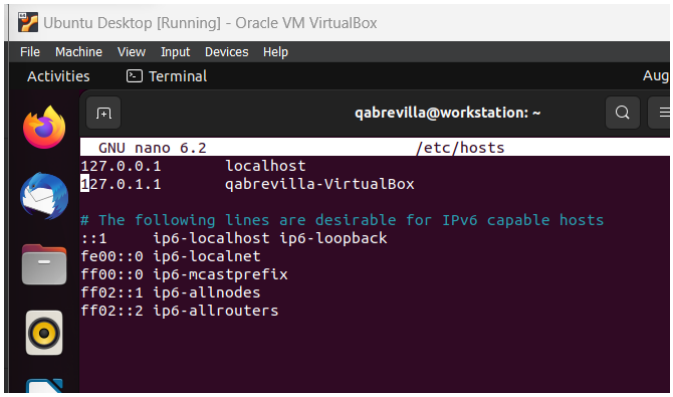


before

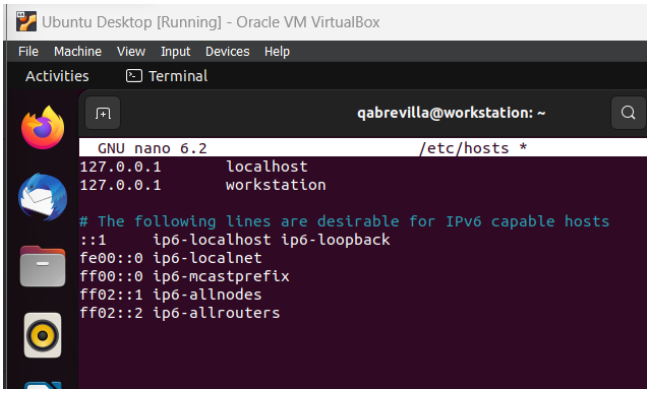


after

2.3 Type 127.0.0.1 workstation for the Local Machine



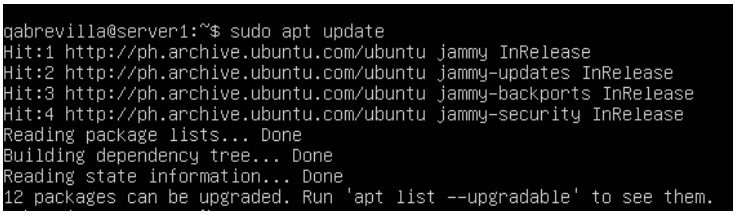
before



after

Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.



```
qabrevilla@server2:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
qabrevilla@workstation:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
77 packages can be upgraded. Run 'apt list --upgradable' to see them.
qabrevilla@workstation:~$
```

sudo update

```
qabrevilla@server1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils cloud-init git git-man initramfs-tools initramf
  libapt-pkg6.0 libldap-2.5-0 libldap-common sosreport
12 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,747 kB of archives.
After this operation, 838 kB disk space will be freed.
Do you want to continue? [Y/n] _
```

```
qabrevilla@server2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils cloud-init git git-man initramfs-tools initramfs
  libapt-pkg6.0 libldap-2.5-0 libldap-common sosreport
12 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,747 kB of archives.
After this operation, 838 kB disk space will be freed.
Do you want to continue? [Y/n]
```

```
qabrevilla@workstation:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
#
# You can verify the status of security fixes usi
# E.g., a recent Ruby vulnerability can be checke
```

sudo upgrade

2. Install the SSH server using the command *sudo apt install openssh-server*.

```
qabrevilla@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
qabrevilla@server1:~$
```

```
qabrevilla@server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
qabrevilla@server2:~$ _
```

```
qabrevilla@workstation:~$ sudo apt install openssh-server
[sudo] password for qabrevilla:
```

3. Verify if the SSH service has started by issuing the following commands:

3.1 sudo service ssh start

```
qabrevilla@server1:~$ sudo service ssh start
```

```
qabrevilla@server2:~$ sudo service ssh start
```

```
qabrevilla@workstation:~$ sudo service ssh start
qabrevilla@workstation:~$
```

3.2 sudo systemctl status ssh

```
qabrevilla@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-22 14:33:00 UTC; 48min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 699 (sshd)
      Tasks: 1 (limit: 5440)
     Memory: 4.4M
        CPU: 69ms
     CGroup: /system.slice/ssh.service
             └─699 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 14:33:00 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 14:33:00 server1 sshd[699]: Server listening on 0.0.0.0 port 22.
Aug 22 14:33:00 server1 sshd[699]: Server listening on :: port 22.
Aug 22 14:33:00 server1 systemd[1]: Started OpenBSD Secure Shell server.
qabrevilla@server1:~$
```

```
qabrevilla@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-22 14:43:21 UTC; 39min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 705 (sshd)
      Tasks: 1 (limit: 5395)
     Memory: 4.4M
        CPU: 27ms
     CGroup: /system.slice/ssh.service
             └─705 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 14:43:21 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 14:43:21 server2 sshd[705]: Server listening on 0.0.0.0 port 22.
Aug 22 14:43:21 server2 sshd[705]: Server listening on :: port 22.
Aug 22 14:43:21 server2 systemd[1]: Started OpenBSD Secure Shell server.
qabrevilla@server2:~$
```

```
qabrevilla@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pre
     Active: active (running) since Tue 2023-08-22 23:30:12 PST; 35s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 32244 (sshd)
      Tasks: 1 (limit: 7255)
     Memory: 1.7M
        CPU: 19ms
     CGroup: /system.slice/ssh.service
             └─32244 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 star

Aug 22 23:30:12 workstation systemd[1]: Starting OpenBSD Secure Shell ser
Aug 22 23:30:12 workstation sshd[32244]: Server listening on 0.0.0.0 port
Aug 22 23:30:12 workstation sshd[32244]: Server listening on :: port 22.
Aug 22 23:30:12 workstation systemd[1]: Started OpenBSD Secure Shell serv
lines 1-16/16 (END)
```

4. Configure the firewall to all port 22 by issuing the following commands:

4.1 sudo ufw allow ssh

```
qabrevilla@server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
qabrevilla@server1:~$
```

```
qabrevilla@server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

```
qabrevilla@workstation:~$ sudo ufw allow ssh
[sudo] password for qabrevilla:
Sorry, try again.
[sudo] password for qabrevilla:
Rules updated
Rules updated (v6)
qabrevilla@workstation:~$
```

4.2 sudo ufw enable

```
qabrevilla@server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
qabrevilla@server1:~$ _
```

```
qabrevilla@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

```
qabrevilla@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
qabrevilla@workstation:~$
```

4.3 sudo ufw status

```
qabrevilla@server1:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)

qabrevilla@server1:~$
```

```
qabrevilla@server2:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)

qabrevilla@server2:~$
```

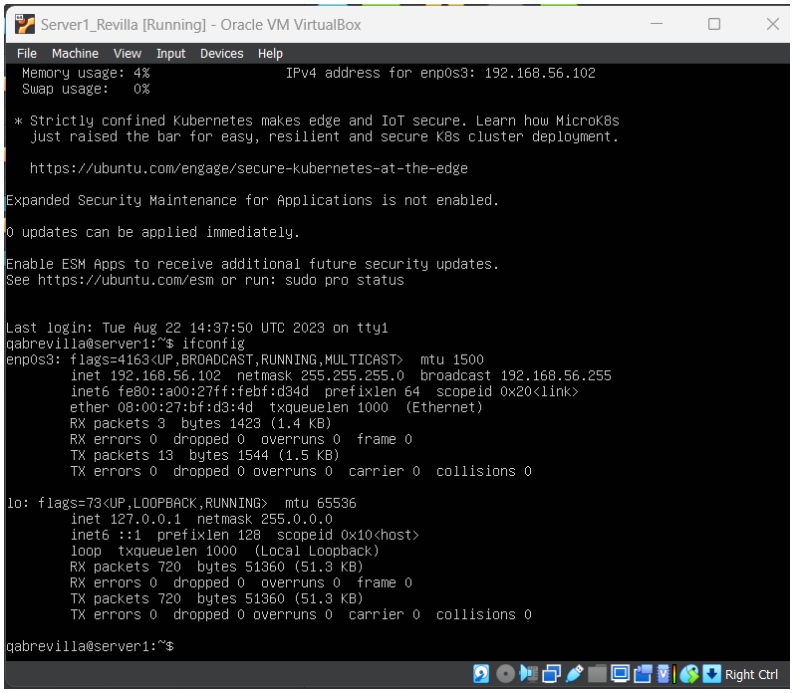
```
qabrevilla@workstation:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)

qabrevilla@workstation:~$
```

Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

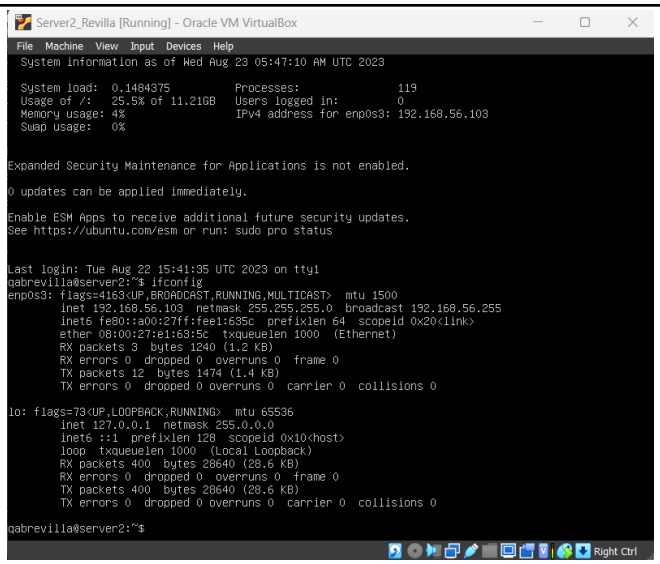
```
qabrevilla@server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:febf:d34d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bf:d3:4d  txqueuelen 1000  (Ethernet)
        RX packets 5790  bytes 8353995 (8.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1768  bytes 115217 (115.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 128  bytes 11332 (11.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 128  bytes 11332 (11.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

qabrevilla@server1:~$ _
```

1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.

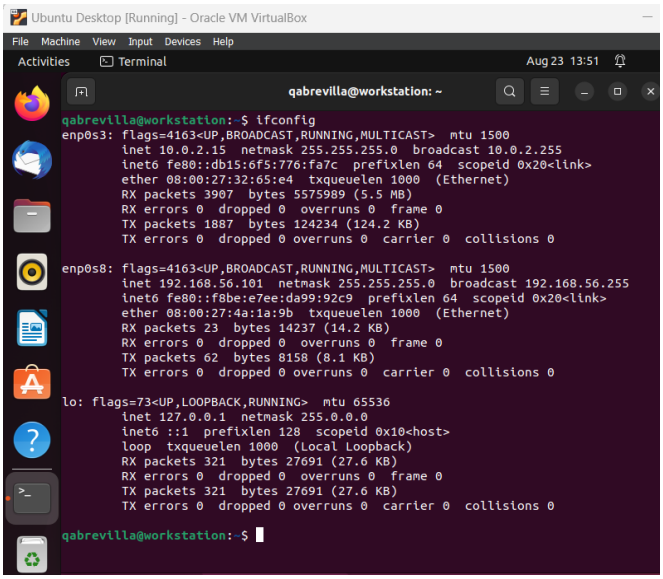
   1.1 Server 1 IP address: 192.168.56.102



   1.2 Server 2 IP address: 192.168.56.103

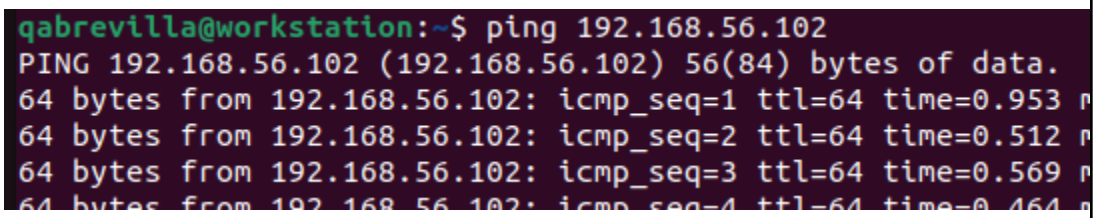


1.3 Server 3 IP address: 192.168.56.101



2. Make sure that they can ping each other.

2.1 Connectivity test for Local Machine 1 to Server 1: ☐ Successful ☐ Not Successful



2.2 Connectivity test for Local Machine 1 to Server 2: ☐ Successful ☐ Not Successful

```
qabrevilla@workstation:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.936 m
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.613 m
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.641 m
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.557 m
^Z
```

2.3 Connectivity test for Server 1 to Server 2: ☐ Successful ☐ Not Successful

```
qabrevilla@server1:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.921 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.560 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.637 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.663 ms
^Z
```

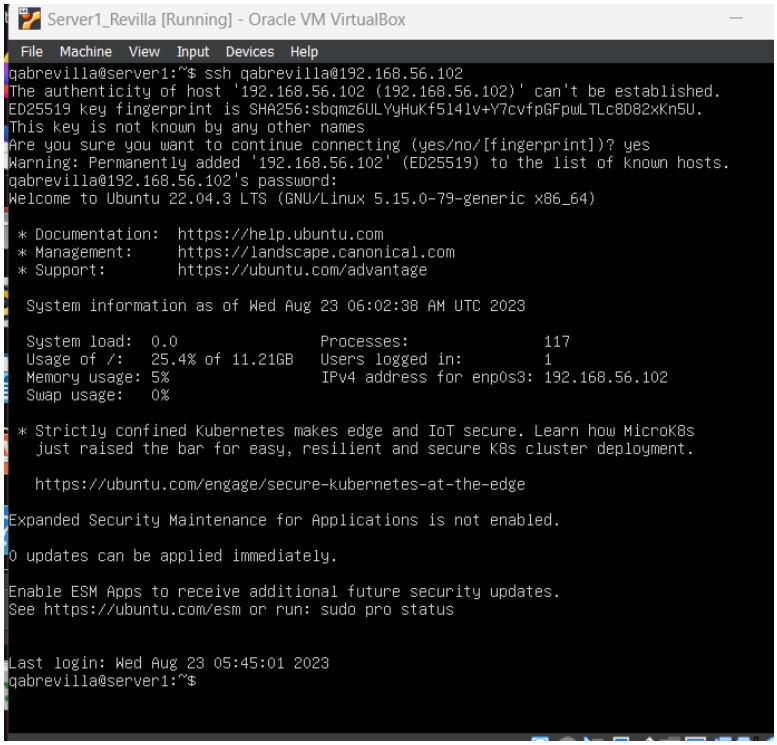
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

1. On the Local Machine, issue the following commands:

1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted

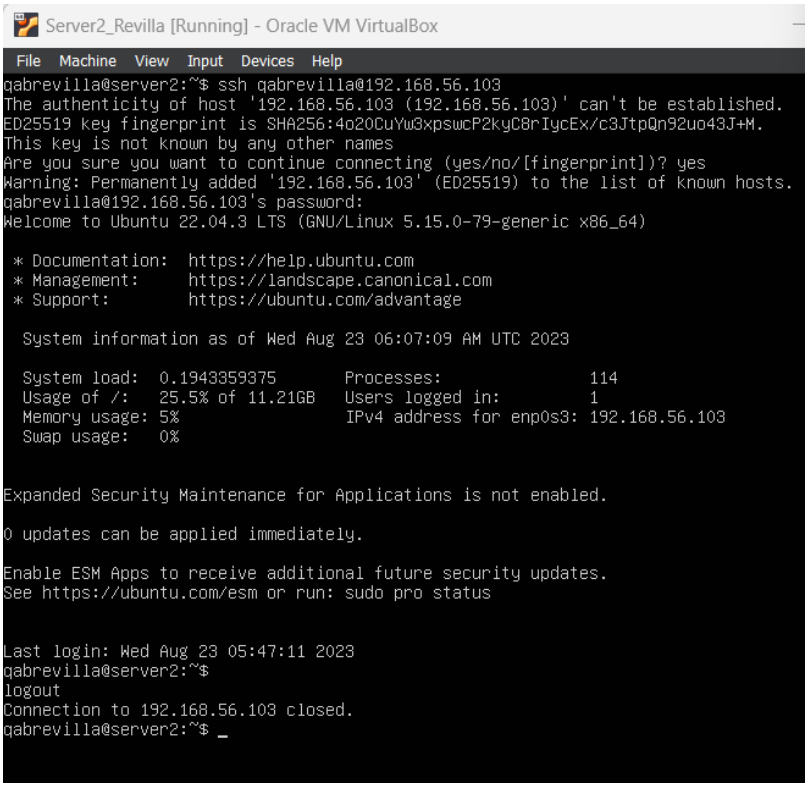
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*



2. Logout of Server 1 by issuing the command *control + D.*

```
qabrevilla@server1:~$
logout
Connection to 192.168.56.102 closed.
qabrevilla@server1:~$ _
```

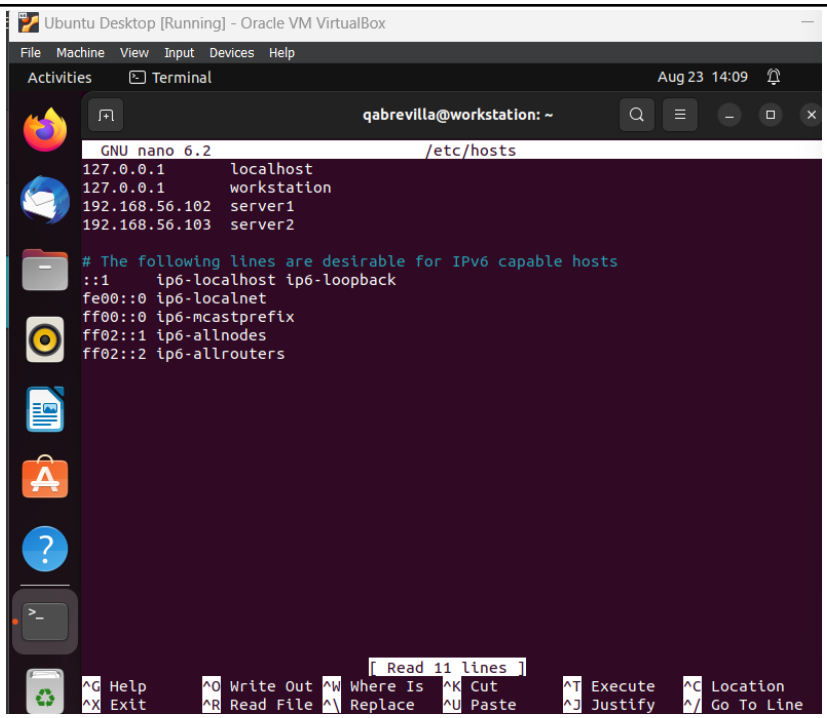
3. Do the same for Server 2.



4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

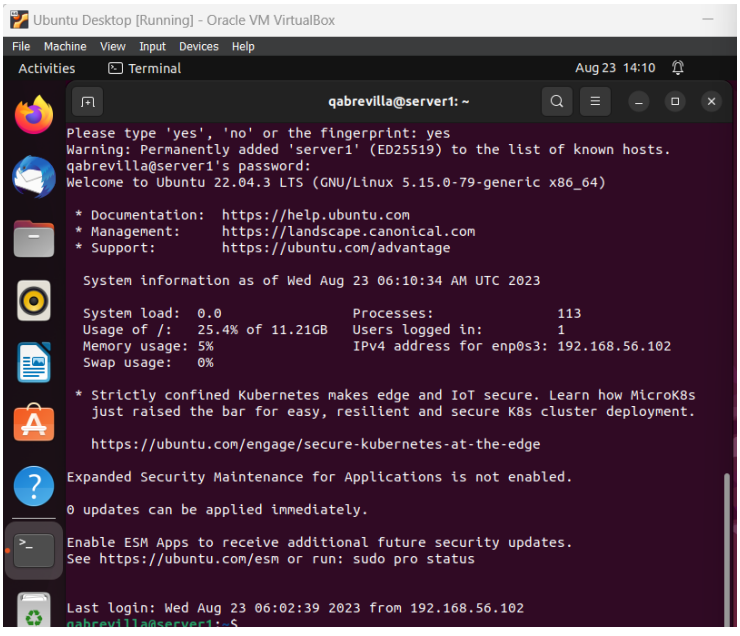
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

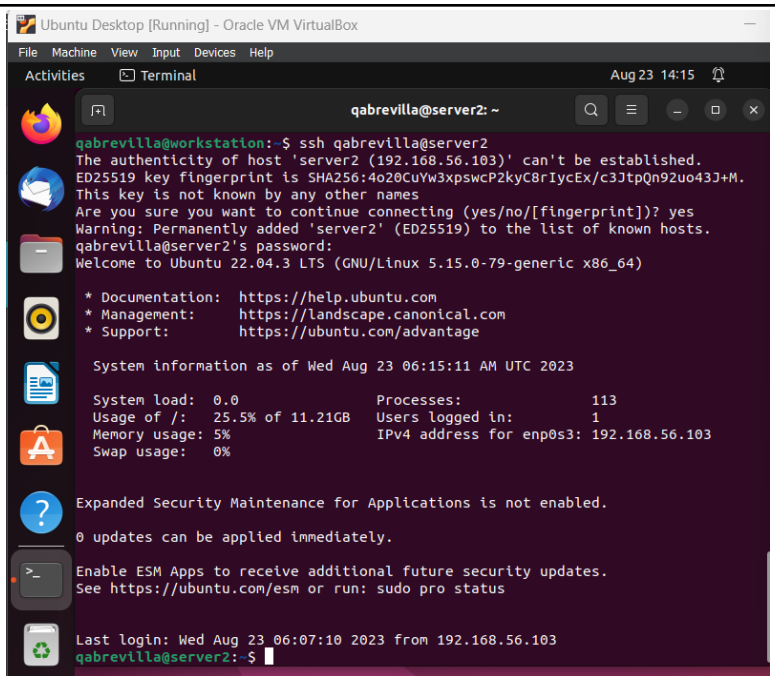
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)



4.3 Save the file and exit.

5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.





Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   In this activity, we're able to use hostname in SSH commands by simply editing the /etc/hosts in the local machine. There, we can set the Ip addresses of the servers and provide the corresponding hostnames. After the configuration, we can now use the SSH commands using hostnames (ex. ssh qabrevilla@server1).

2. How secured is SSH?

   Secure Socket Shell (SSH) is a network communication protocol that enables the network between two devices. It also comes with a strong encryption method using password authentication for safe remote access and communication. In the activity, I understand the methods of accessing and managing servers and computers and how it will be a good skill for computer engineers.

Conclusion:

From this activity, We were able to experience connecting ubuntu desktop and servers in a virtual machine. We used ssh commands to configure the connection between the devices. I also learned how important ip addresses play an important role in

connecting computers. We change /etc/hosts and /etc/hostname to configure the name and ip addresses of the devices for it to connect the system.