Name: Carag, Carl Jervie B.	**Date Performed: 08/08/25**
Course/Section: BSCPE CPE31S2	**Date Submitted: 08/08/25**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Semester**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

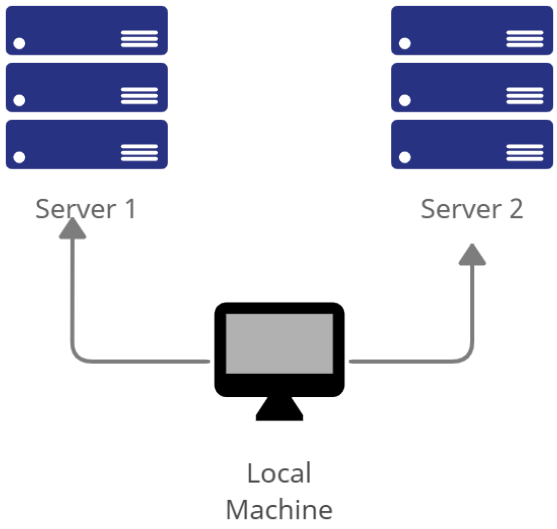
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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
   1.2 Use server2 for Server 2
   1.3 Use workstation for the Local Machine
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
2.2 Type 127.0.0.1 server 2 for Server 2
2.3 Type 127.0.0.1 workstation for the Local Machine

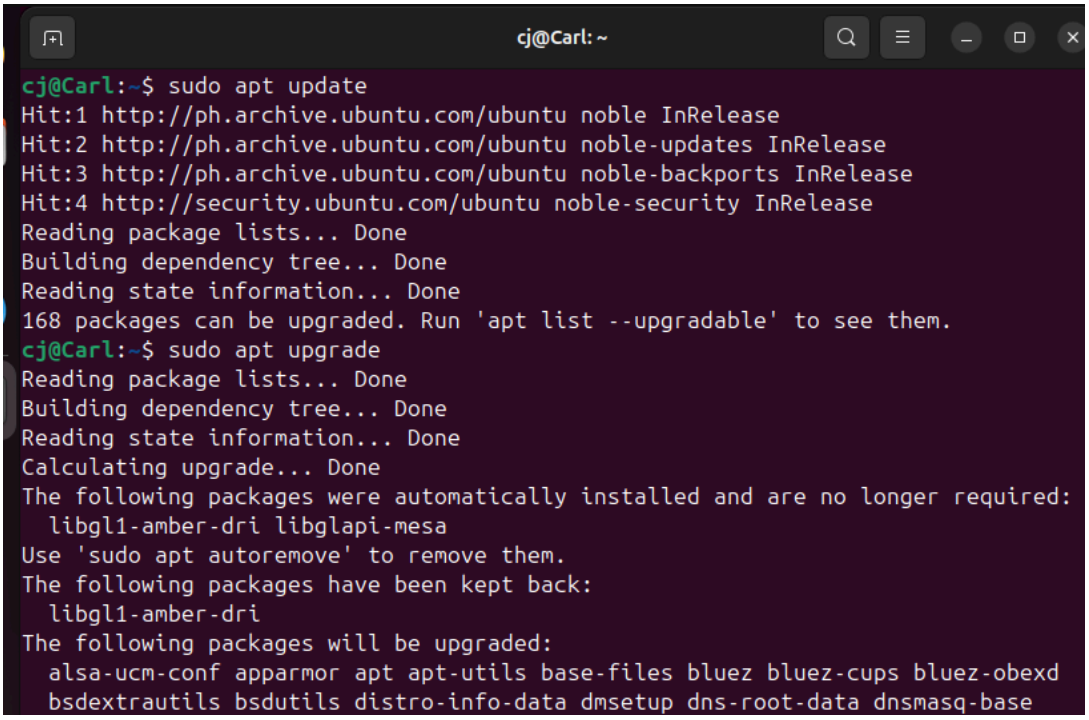
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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

Server 2

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

Server 1



Workstation

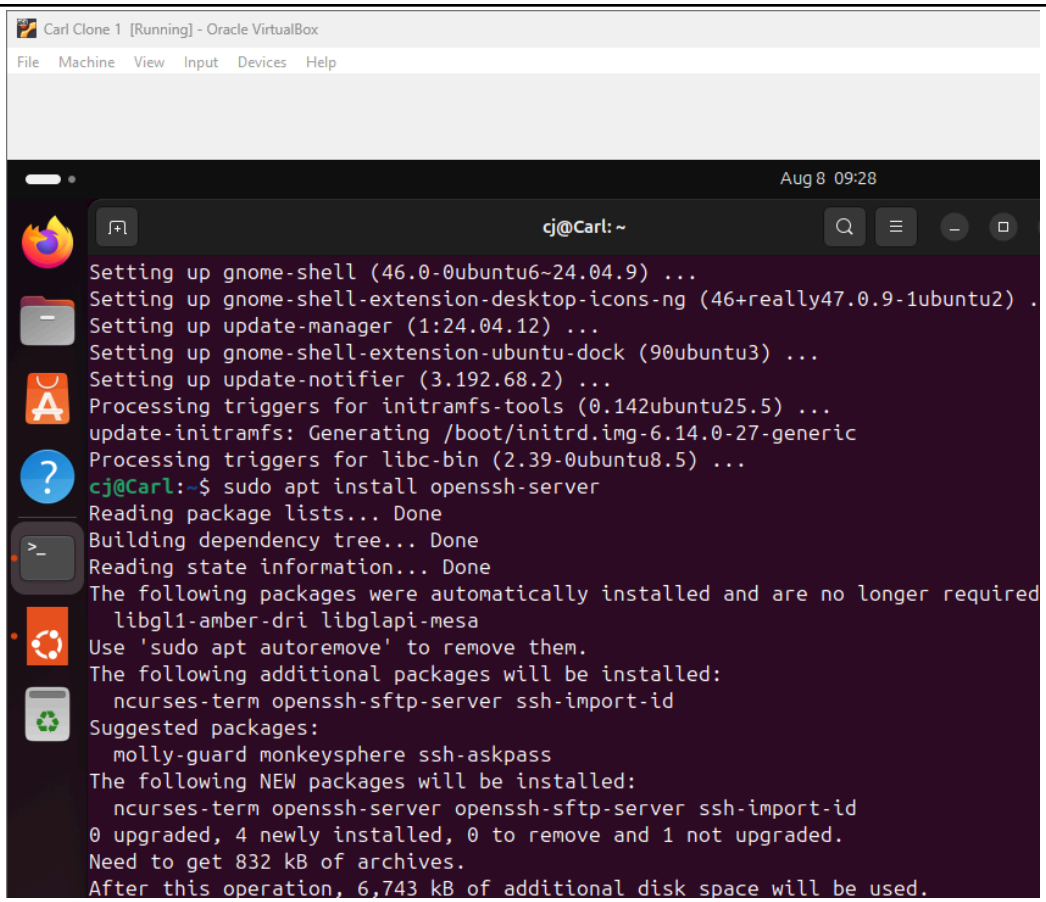
```
cj@Carl:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
168 packages can be upgraded. Run 'apt list --upgradable' to see them.
cj@Carl:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-mesa
```

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

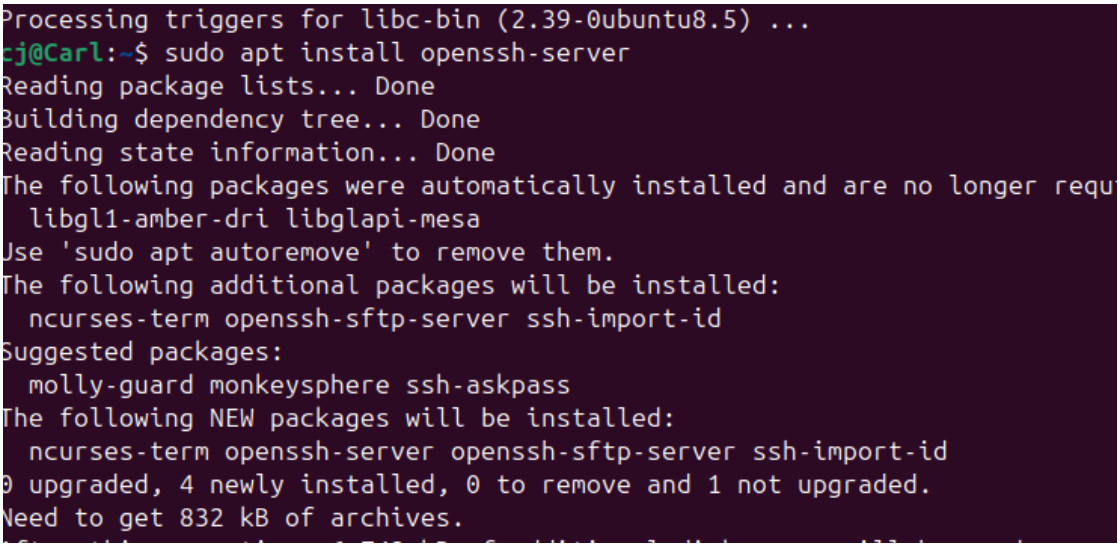
Workstation

```
Processing triggers for libc-bin (2.39-0ubuntu8.5) ...
cj@Carl:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longe
  libgl1-amber-dri libglapi-mesa
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
```

Server 1

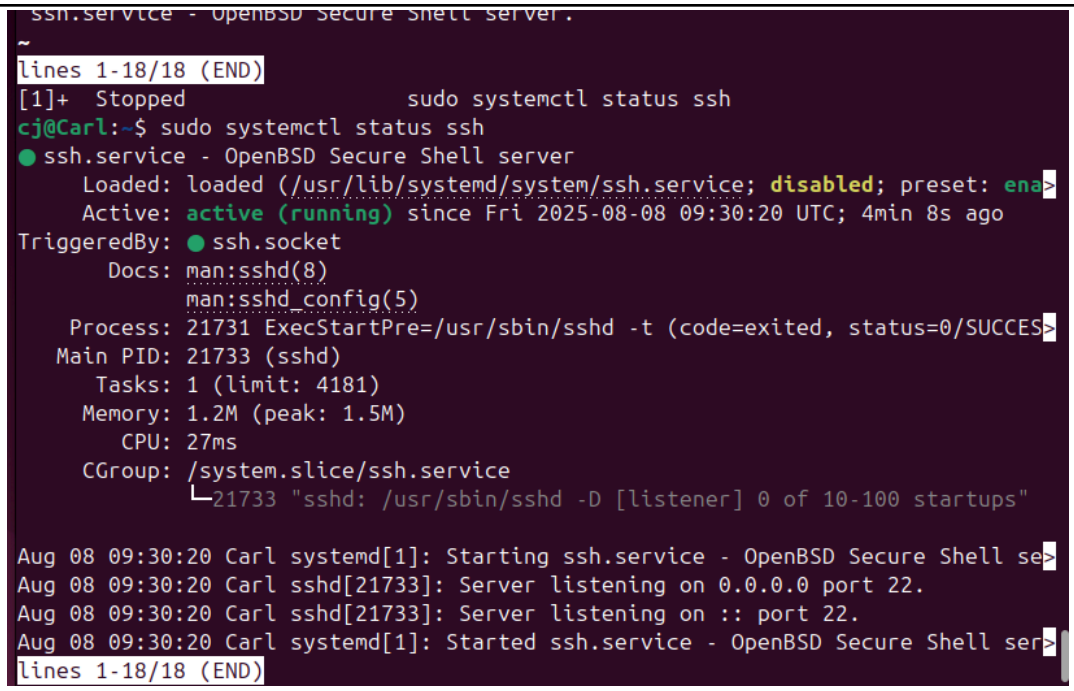


Server 2

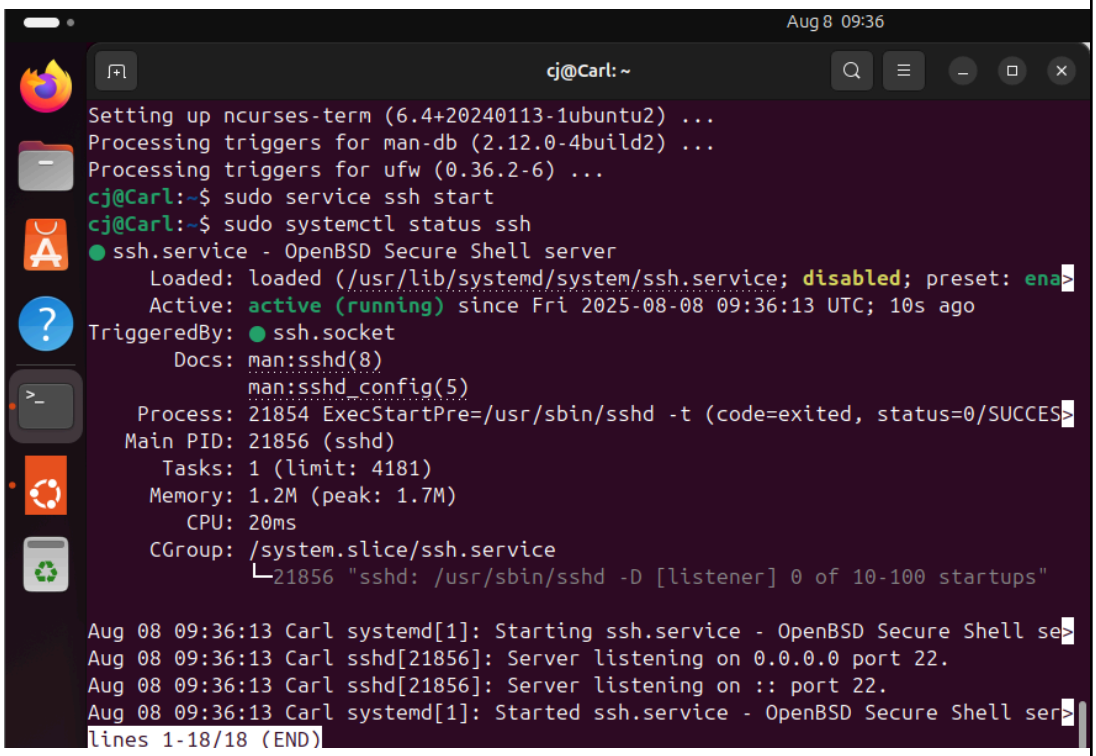


3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*

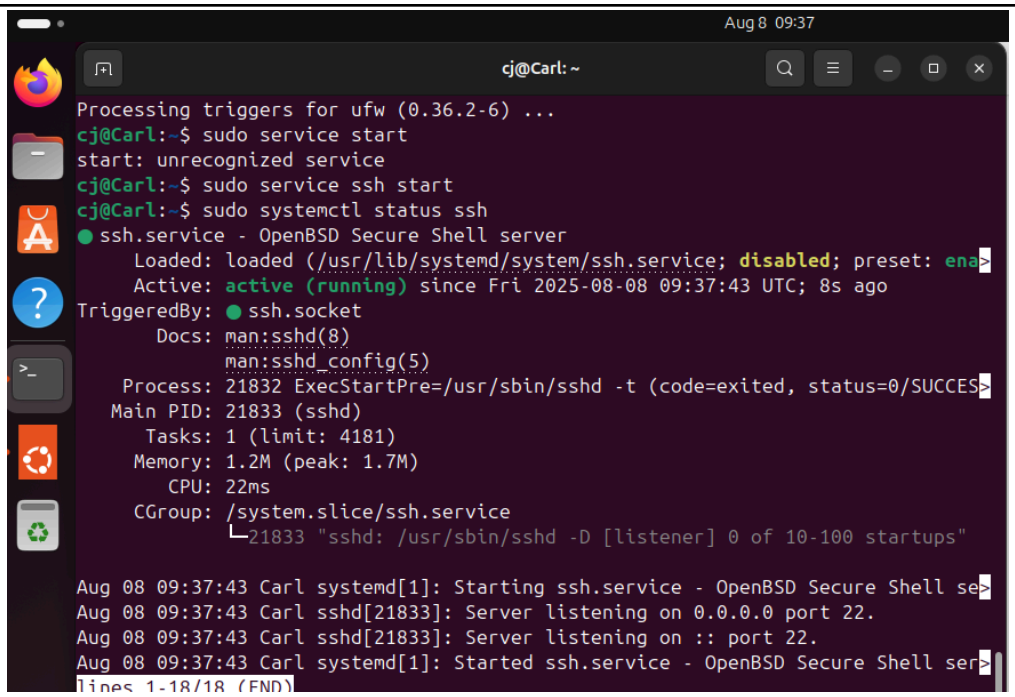
Server 2



Server 1



Workstation



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

4.1 *sudo ufw allow ssh*

Workstation

```
cj@Carl:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
cj@Carl:~$
```

Server 1

```
cj@Carl:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
cj@Carl:~$
```

Server 2

```
cj@Carl:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
cj@Carl:~$
```

4.2 *sudo ufw enable*

Workstation

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

Server 1

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

Server 2

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

4.3 *sudo ufw status*

Workstation

```
cj@Carl:~$ sudo ufw status
Status: active

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

cj@Carl:~$
```

Server 1

```
cj@Carl:~$ sudo ufw status
Status: active

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

cj@Carl:~$
```

Server 2

```
cj@Carl:~$ sudo ufw status
Status: active

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

cj@Carl:~$
```

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

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. 106

Server 1

```
cj@Carl:~$ 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 fd00::fb63:321:8595:1583  prefixlen 64  scopeid 0x0<glob
        inet6 fd00::a00:27ff:fe91:5a72  prefixlen 64  scopeid 0x0<glob
        inet6 fe80::a00:27ff:fe91:5a72  prefixlen 64  scopeid 0x20<lir
        ether 08:00:27:91:5a:72  txqueuelen 1000  (Ethernet)
        RX packets 155478  bytes 148537958 (148.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 54807  bytes 3339176 (3.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.106  netmask 255.255.255.0  broadcast 192.168.
        inet6 fe80::4d34:db19:5efb:7e06  prefixlen 64  scopeid 0x20<li
        ether 08:00:27:98:2b:e3  txqueuelen 1000  (Ethernet)
        RX packets 420  bytes 74040 (74.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 86  bytes 11099 (11.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

1.2 Server 2 IP address: 192.168.56. 107

Server 2

```
cj@Carl:~$ 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:fe2c:653a  prefixlen 64  scopeid 0x20<link>
        inet6 fd00::39ce:a625:f5e5:d4d9  prefixlen 64  scopeid 0x0<global>
        inet6 fd00::a00:27ff:fe2c:653a  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:2c:65:3a  txqueuelen 1000  (Ethernet)
        RX packets 154053  bytes 148460146 (148.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 55213  bytes 3363508 (3.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.107  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::2a0f:a099:4ab7:50d6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:68:fd:5f  txqueuelen 1000  (Ethernet)
        RX packets 320  bytes 59272 (59.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93  bytes 12285 (12.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

1.3 Server 3 IP address: 127.0.0.1

Workstation

```
cj@Carl:~$ 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:fe08:1c78  prefixlen 64  scopeid 0x20<link>
        inet6 fd00::f2e9:c8f3:a3d9:4f3d  prefixlen 64  scopeid 0x0<global
        inet6 fd00::a00:27ff:fe08:1c78  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:08:1c:78  txqueuelen 1000  (Ethernet)
        RX packets 153053  bytes 148408755 (148.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 51449  bytes 3138196 (3.1 MB)
        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 192  bytes 19517 (19.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 192  bytes 19517 (19.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Make sure that they can ping each other.

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

```
cj@Workstation:~$ ping 192.168.56.106
PING 192.168.56.106 (192.168.56.106) 56(84) bytes of data.
64 bytes from 192.168.56.106: icmp_seq=1 ttl=64 time=1.67 ms
64 bytes from 192.168.56.106: icmp_seq=2 ttl=64 time=0.440 ms
64 bytes from 192.168.56.106: icmp_seq=3 ttl=64 time=0.944 ms
64 bytes from 192.168.56.106: icmp_seq=4 ttl=64 time=0.489 ms
64 bytes from 192.168.56.106: icmp_seq=5 ttl=64 time=0.881 ms
64 bytes from 192.168.56.106: icmp_seq=6 ttl=64 time=0.894 ms
64 bytes from 192.168.56.106: icmp_seq=7 ttl=64 time=0.807 ms
^Z
[1]+  Stopped                 ping 192.168.56.106
cj@Workstation:~$
```

Successful

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

```
cj@Workstation:~$ ping 192.168.56.107
PING 192.168.56.107 (192.168.56.107) 56(84) bytes of data.
64 bytes from 192.168.56.107: icmp_seq=1 ttl=64 time=3.35 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.665 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=0.656 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=0.829 ms
64 bytes from 192.168.56.107: icmp_seq=5 ttl=64 time=0.598 ms
64 bytes from 192.168.56.107: icmp_seq=6 ttl=64 time=0.598 ms
^Z
[2]+  Stopped                 ping 192.168.56.107
cj@Workstation:~$
```

Successful

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

```
cj@Server1:~$ ping 192.168.56.107
PING 192.168.56.107 (192.168.56.107) 56(84) bytes of data.
ping: Warning: time of day goes back (-1 s), taking countermeasures
64 bytes from 192.168.56.107: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.657 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=0.607 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=0.689 ms
64 bytes from 192.168.56.107: icmp_seq=5 ttl=64 time=0.657 ms
64 bytes from 192.168.56.107: icmp_seq=6 ttl=64 time=0.564 ms
^Z
[1]+  Stopped                 ping 192.168.56.107
cj@Server1:~$
```

Successful

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

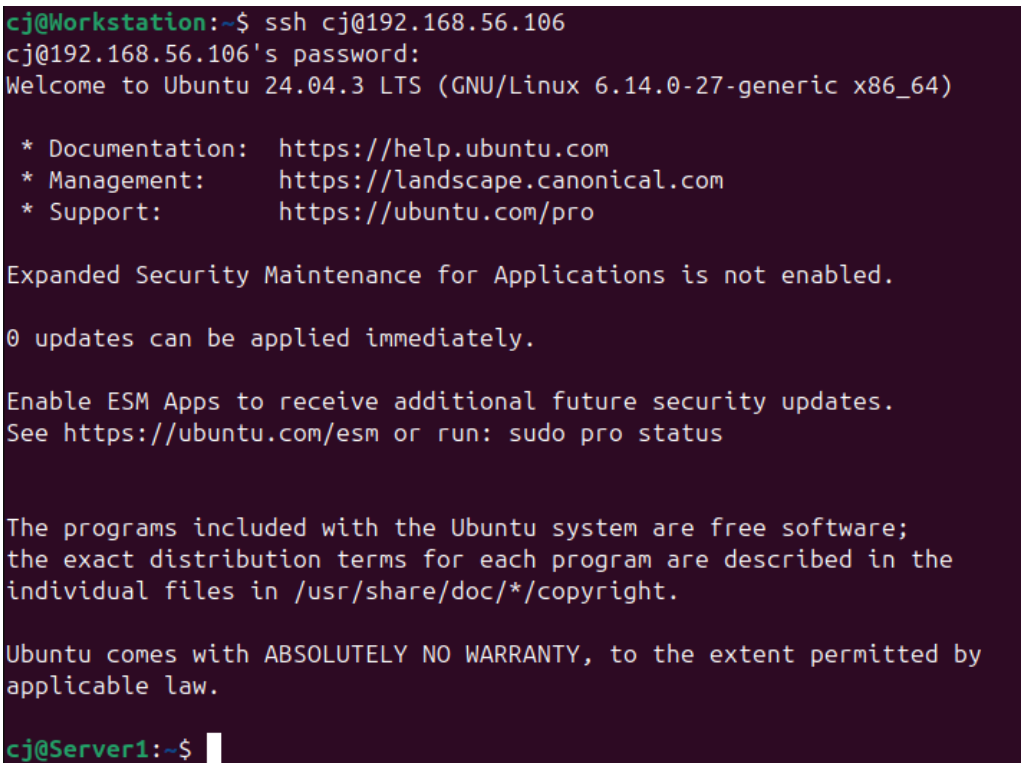
1. On the Local Machine, issue the following commands:

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

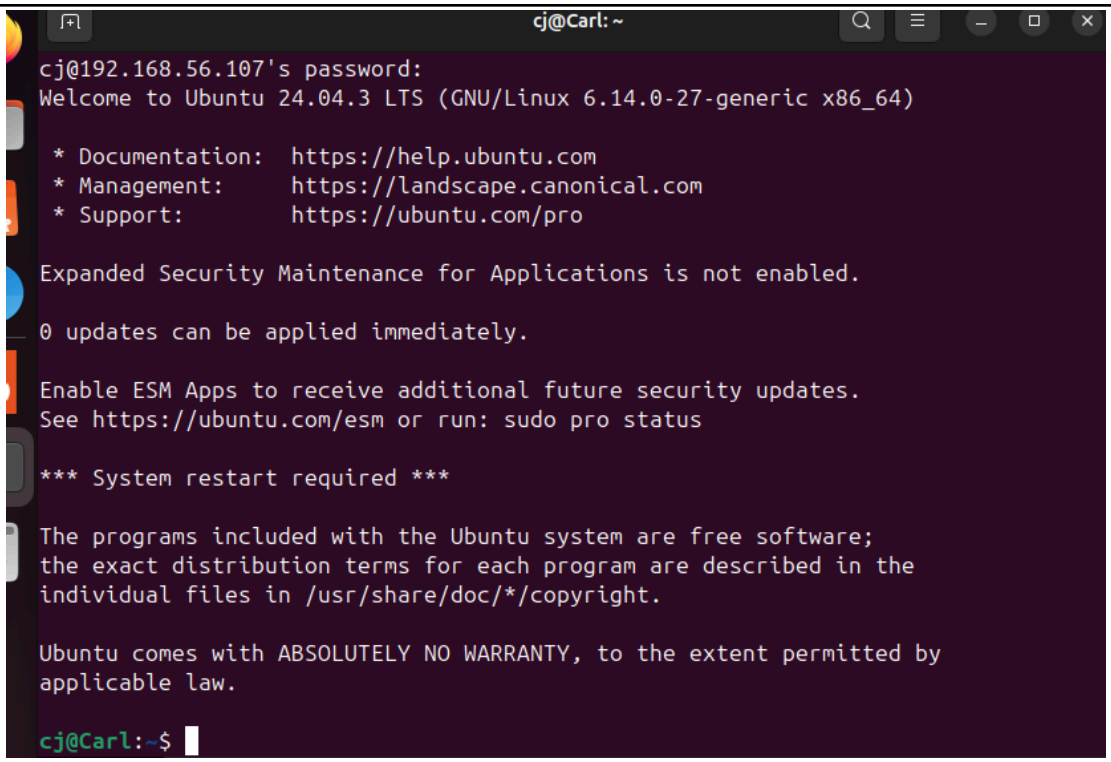
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

Server 1



2. Logout of Server 1 by issuing the command *control + D.*
3. Do the same for Server 2.

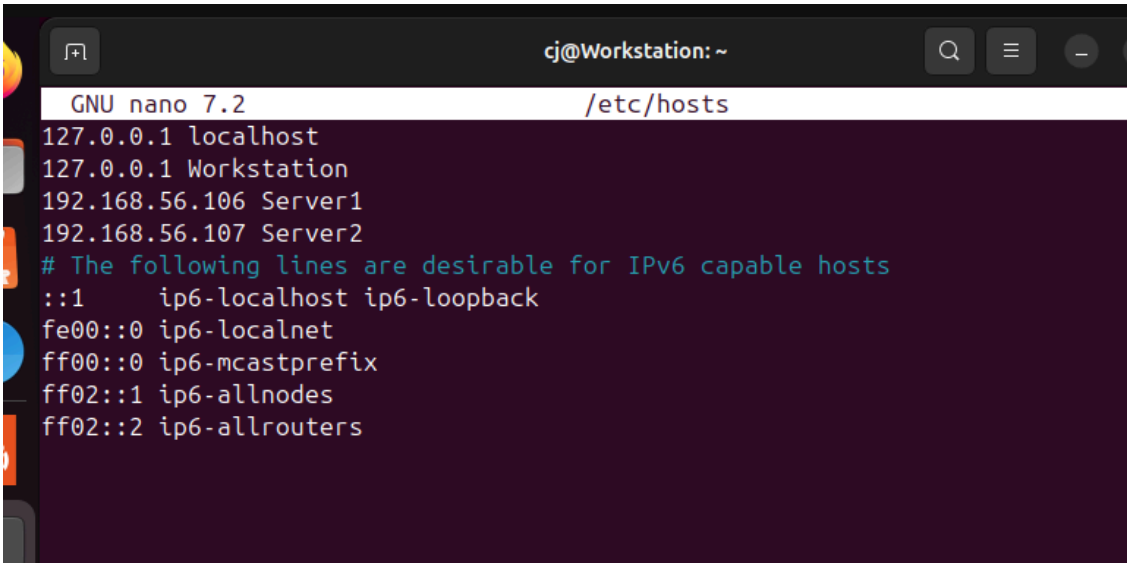
Server 2



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)

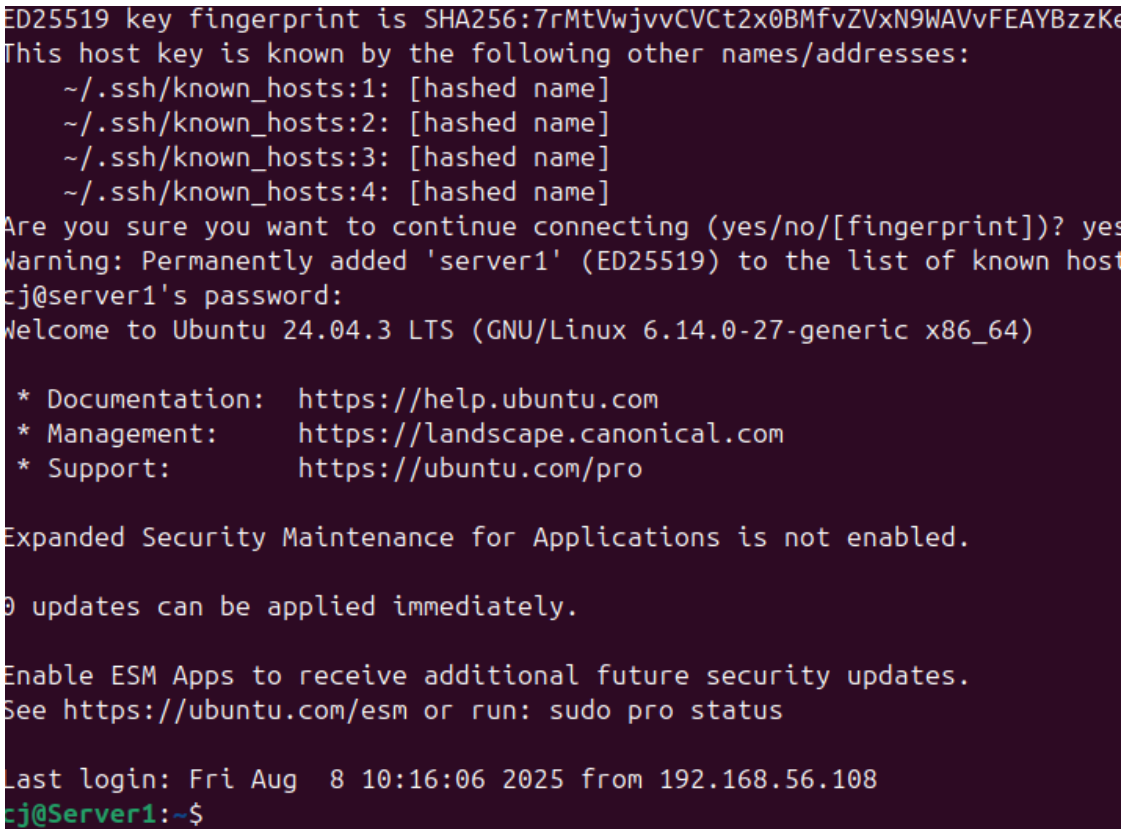


4.3 Save the file and exit.

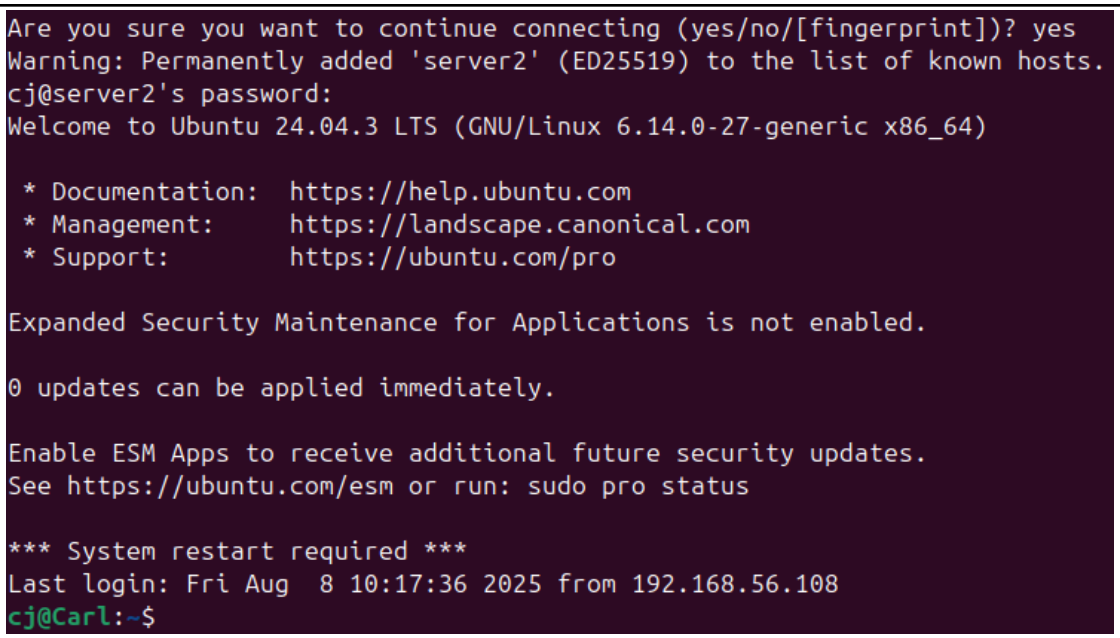
5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example,

try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1.

Server 1



Server 2



6. 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?
   - The OS may go first with the hosts file, and when the hostname isn't found on the local host file it may consult the Domain Name System.
2. How secured is SSH?
   - The ssh encrypts all the data being transmitted between the client and server.