Name: Cruz, Patrick Danielle C.	**Date Performed: Aug 8, 2025**
Course/Section: CPE212 - CPE31S4	**Date Submitted: Aug 8, 2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st semester**

Activity 1: Configure Network using Virtual Machines

1. Objectives:

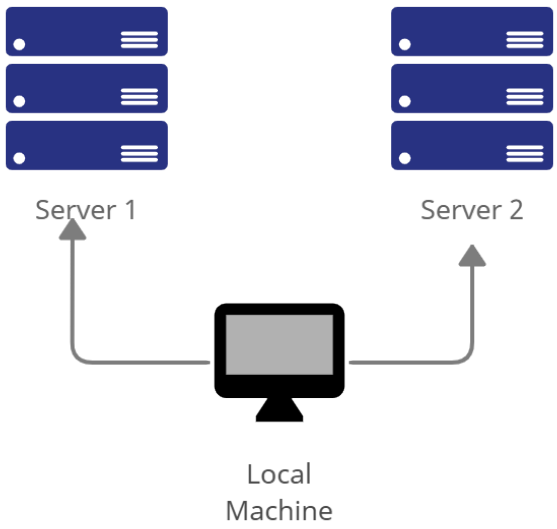
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

1.2. Set-up a Virtual Network and Test Connectivity of VMs

2. Discussion:

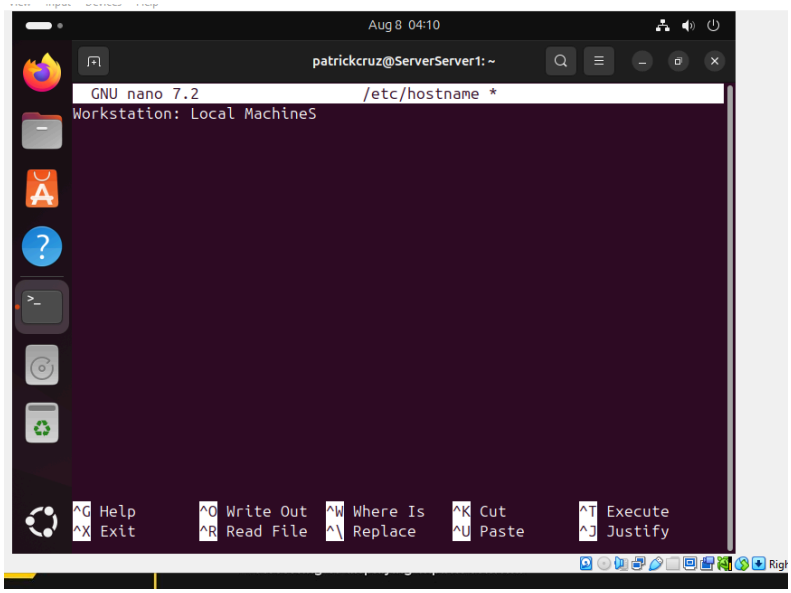
Network Topology:

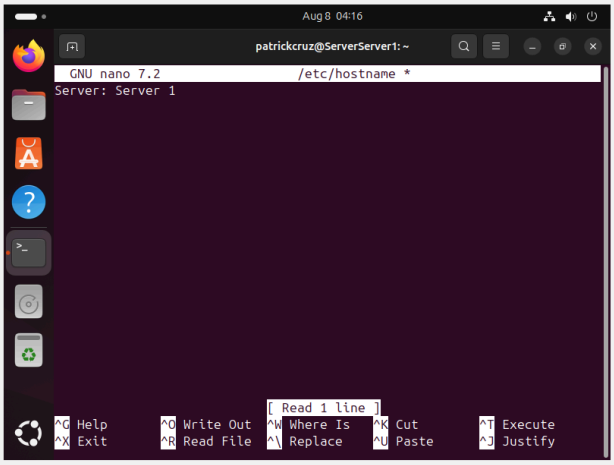
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).

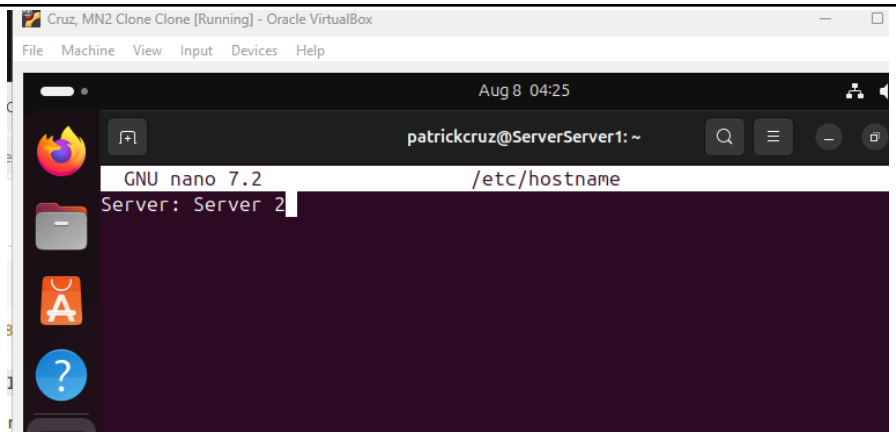


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

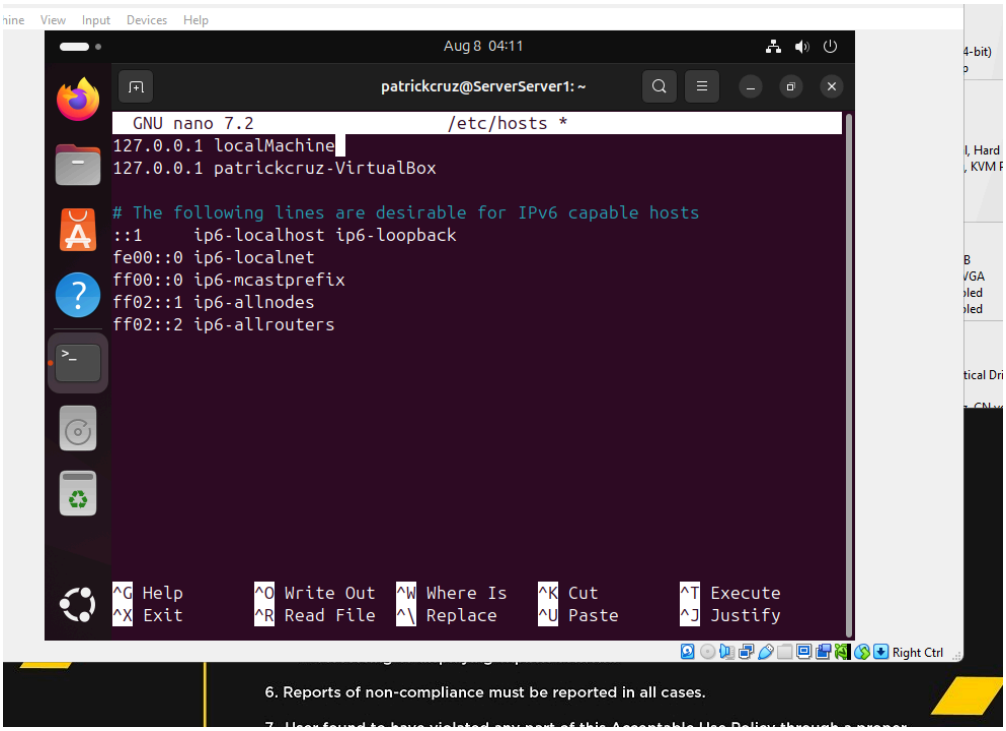
1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1
   1.2 Use server2 for Server 2
   1.3 Use workstation for the Local Machine

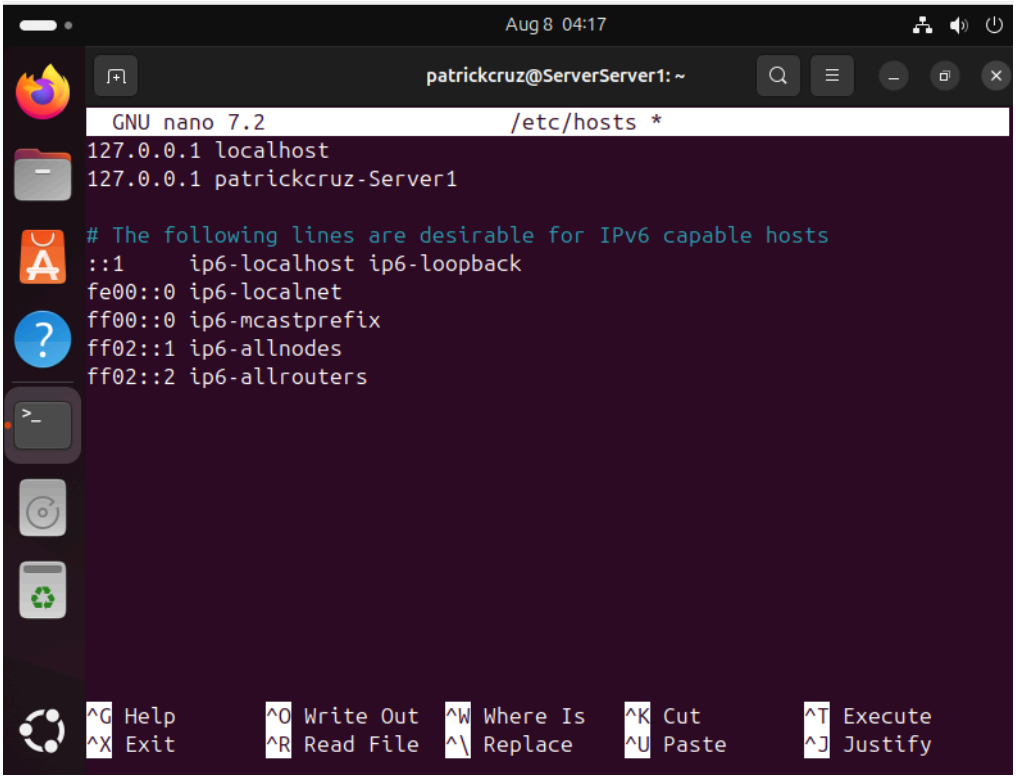

Aug 8 04:10
patrickcruz@ServerServer1: ~
GNU nano 7.2 /etc/hostname *
Workstation: Local MachineS
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

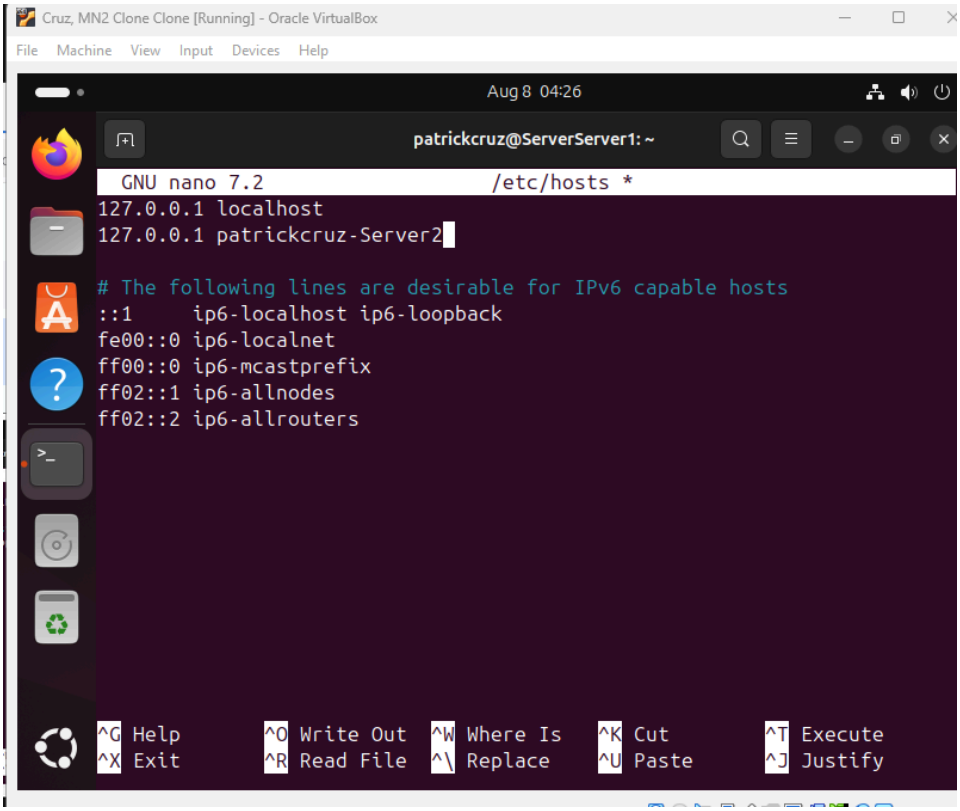

Aug 8 04:16
patrickcruz@ServerServer1: ~
GNU nano 7.2 /etc/hostname *
Server: Server 1
[Read 1 line]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify



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

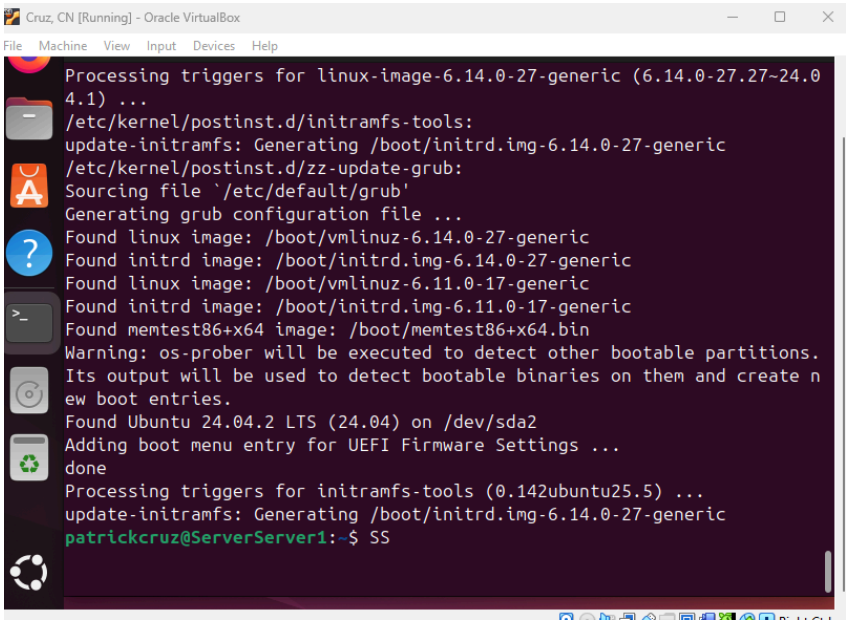



Aug 8 04:17
patrickcruz@ServerServer1: ~
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 patrickcruz-Server1
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

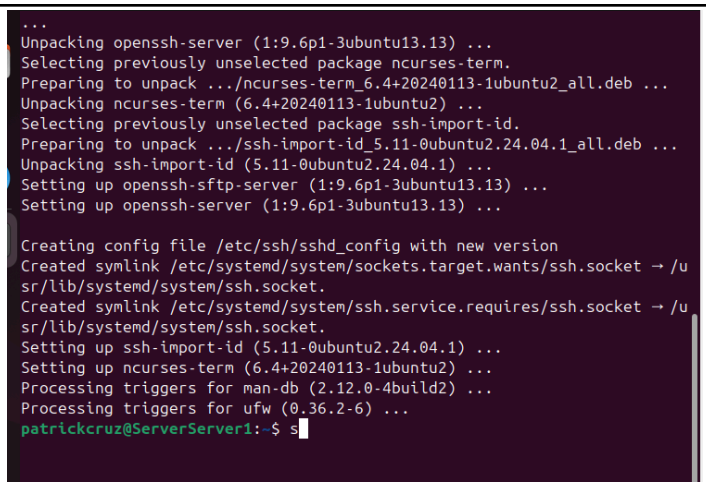

Cruz, MN2 Clone Clone [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 04:26
patrickcruz@ServerServer1: ~
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 patrickcruz-Server2
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

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

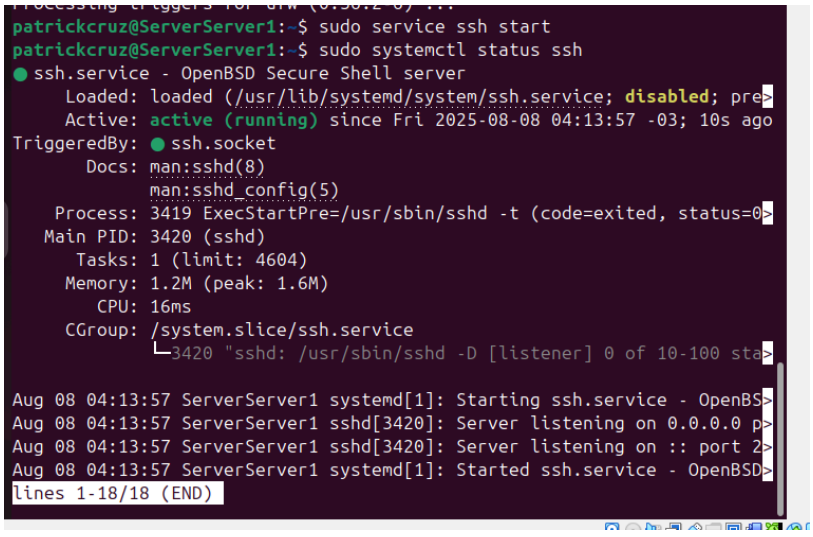
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.



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



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*



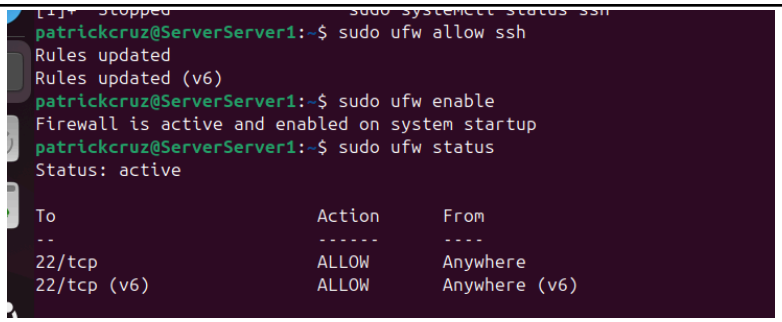
```
patrickcruz@ServerServer1:~$ sudo service ssh start
patrickcruz@ServerServer1:~$ sudo systemctl ssh
Unknown command verb 'ssh', did you mean 'stop'?
patrickcruz@ServerServer1:~$ sudo systemctl status  ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; pre>
     Active: active (running) since Fri 2025-08-08 04:20:11 -03; 17s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3384 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0>
   Main PID: 3386 (sshd)
      Tasks: 1 (limit: 4604)
     Memory: 1.2M (peak: 1.7M)
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             └─3386 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 sta>

Aug 08 04:20:11 ServerServer1 systemd[1]: Starting ssh.service - OpenBS>
Aug 08 04:20:11 ServerServer1 sshd[3386]: Server listening on 0.0.0.0 p>
Aug 08 04:20:11 ServerServer1 sshd[3386]: Server listening on :: port 2>
Aug 08 04:20:11 ServerServer1 systemd[1]: Started ssh.service - OpenBSD>
lines 1-18/18 (END)
```

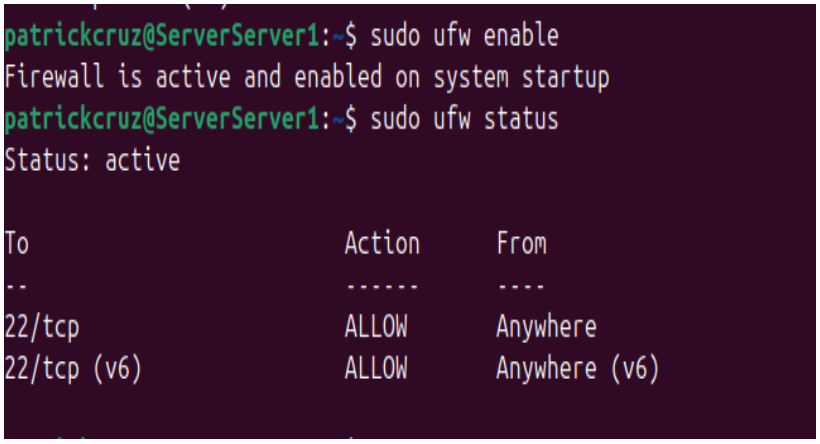
```
patrickcruz@ServerServer1:~$ sudo service ssh start
patrickcruz@ServerServer1:~$ sudo systemctl ssh
Unknown command verb 'ssh', did you mean 'stop'?
patrickcruz@ServerServer1:~$ sudo systemctl status  ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; pre>
     Active: active (running) since Fri 2025-08-08 04:27:33 -03; 14s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3500 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0>
   Main PID: 3501 (sshd)
      Tasks: 1 (limit: 4604)
     Memory: 1.2M (peak: 1.6M)
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─3501 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 sta>

Aug 08 04:27:33 ServerServer1 systemd[1]: Starting ssh.service - OpenBS>
Aug 08 04:27:33 ServerServer1 sshd[3501]: Server listening on 0.0.0.0 p>
Aug 08 04:27:33 ServerServer1 sshd[3501]: Server listening on :: port 2>
Aug 08 04:27:33 ServerServer1 systemd[1]: Started ssh.service - OpenBSD>
lines 1-18/18 (END)
```

4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*


patrickcruz@ServerServer1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
patrickcruz@ServerServer1:~$ sudo ufw enable
Firewall is active and enabled on system startup
patrickcruz@ServerServer1:~$ sudo ufw status
Status: active
To Action From
-- ------ ----
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)


patrickcruz@ServerServer1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
patrickcruz@ServerServer1:~$ sudo ufw enable
Firewall is active and enabled on system startup
patrickcruz@ServerServer1:~$ sudo ufw status
Status: active
To Action From
-- ------ ----
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)


patrickcruz@ServerServer1:~$ sudo ufw enable
Firewall is active and enabled on system startup
patrickcruz@ServerServer1:~$ sudo ufw status
Status: active
To Action From
-- ------ ----
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)

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: 192.168.56.104
   1.2 Server 2 IP address: 192.168.56.107
   1.3 Server 3 IP address: 192.168.56.108

```
patrickcruz@ServerServer1:~$ ping 192.168.56.108 -c 3
PING 192.168.56.108 (192.168.56.108) 56(84) bytes of data.
64 bytes from 192.168.56.108: icmp_seq=1 ttl=64 time=0.589 ms
64 bytes from 192.168.56.108: icmp_seq=2 ttl=64 time=0.377 ms
64 bytes from 192.168.56.108: icmp_seq=3 ttl=64 time=0.433 ms

--- 192.168.56.108 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2075ms
rtt min/avg/max/mdev = 0.377/0.466/0.589/0.089 ms
```

```
patrickcruz@ServerServer1:~$ ping 192.168.56.107 -c 3
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=0.881 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.423 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=0.462 ms

--- 192.168.56.107 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2066ms
rtt min/avg/max/mdev = 0.423/0.588/0.881/0.207 ms
```

```
patrickcruz@ServerServer1:~$ ping 192.168.56.107 -c 4
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=0.442 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.460 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=0.481 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=0.394 ms

--- 192.168.56.107 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3840ms
rtt min/avg/max/mdev = 0.394/0.444/0.481/0.032 ms
```

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

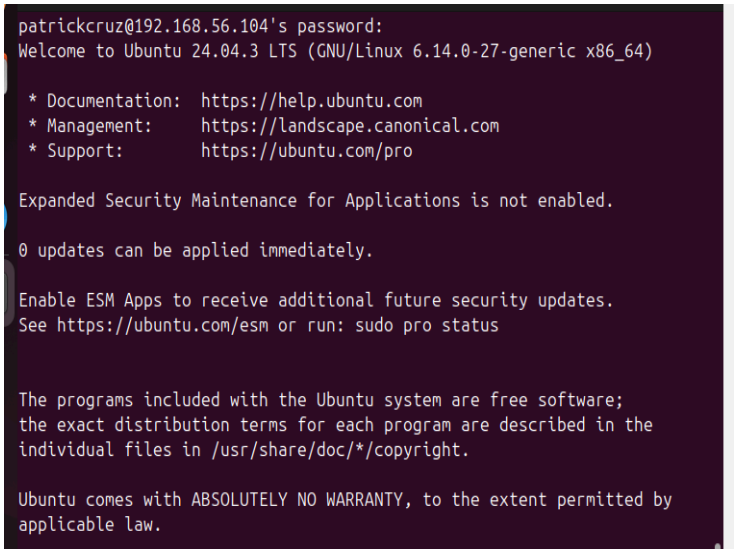
2.3 Connectivity test for Server 1 to Server 2: ✓ Successful ☐ Not Successful

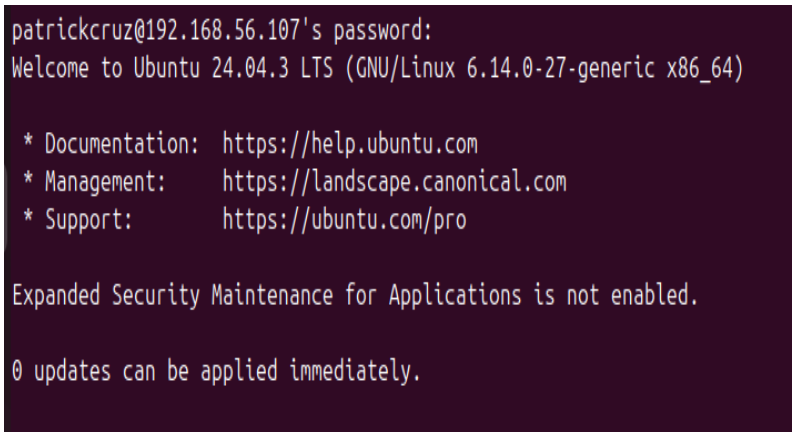
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

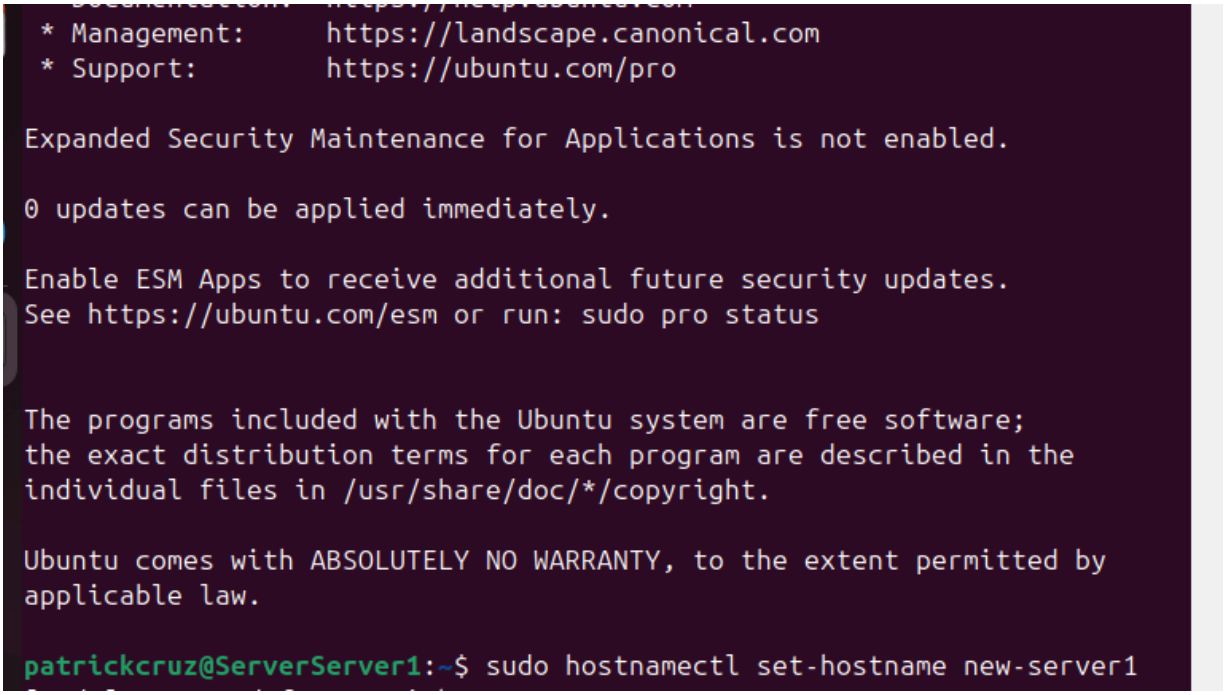
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted

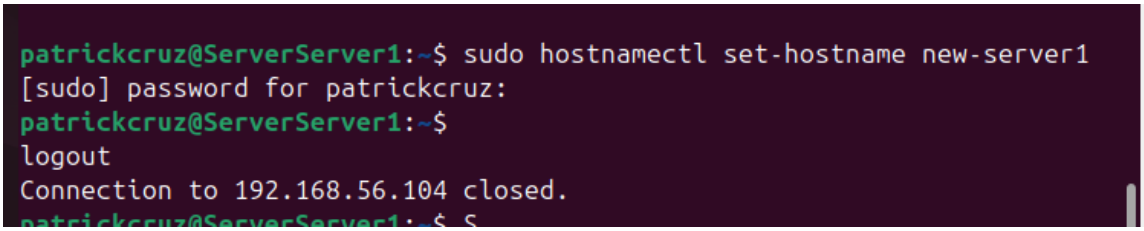




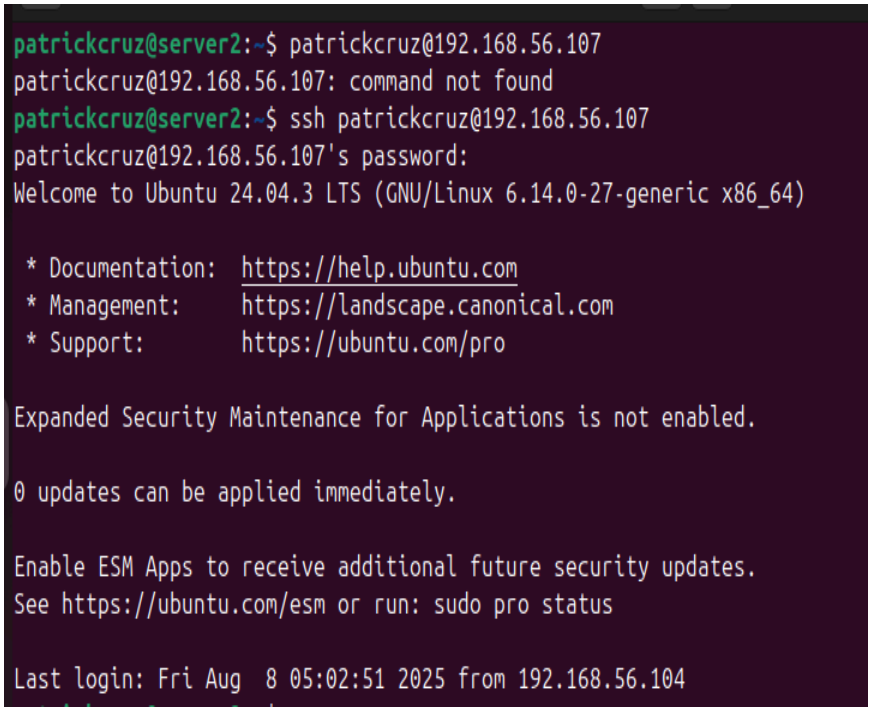
1.3 Verify that you are on server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

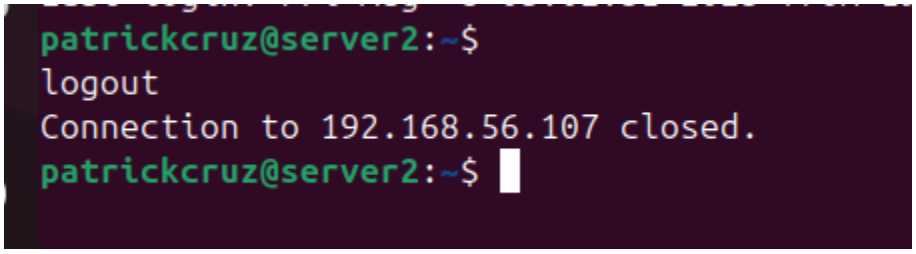


2. Logout of Server 1 by issuing the command *control + D.*



3. Do the same for Server 2.

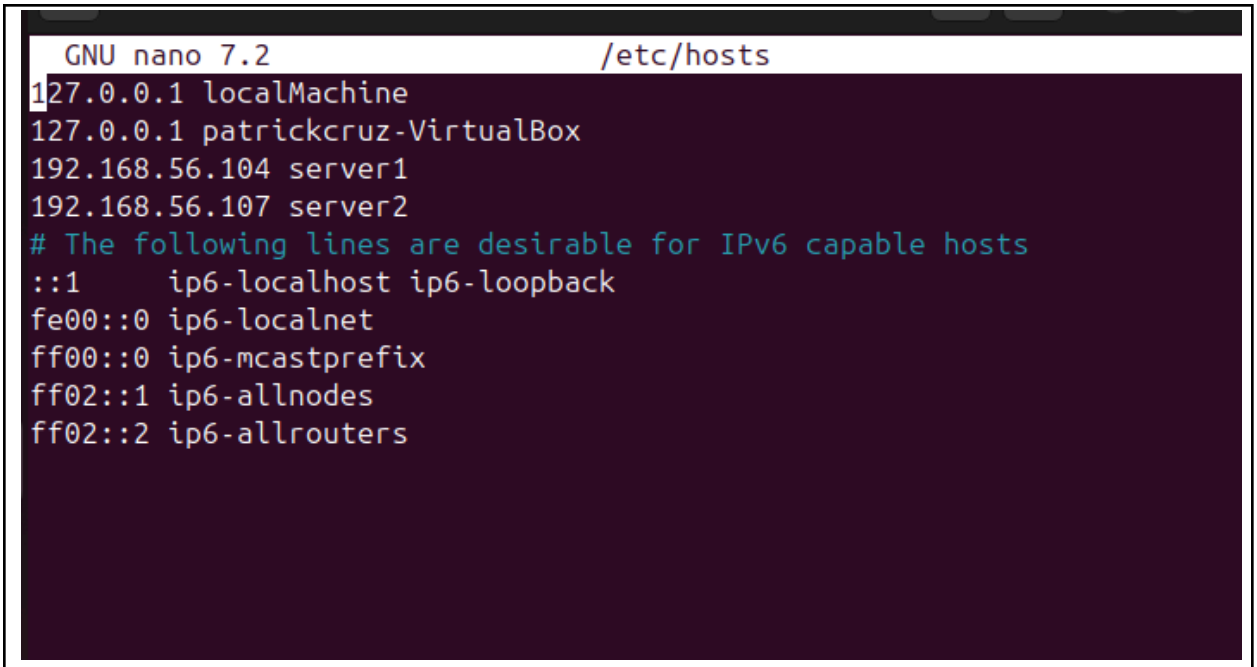




4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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.

```
patrickcruz@ServerServer1:~$ sudo nano /etc/hosts
patrickcruz@ServerServer1:~$ ssh patrickcruz@server1
patrickcruz@server1's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Che
ck your Internet connection or proxy settings

Last login: Fri Aug  8 05:12:55 2025 from 192.168.56.104
```

```
vboxuser@LocalMachine:~$ ssh vboxuser@server2
vboxuser@server2's password:
Welcome to Ubuntu 25.04 (GNU/Linux 6.14.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

0 updates can be applied immediately.

Last login: Fri Aug  8 08:07:47 2025 from 192.168.56.104
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?

- In SSH commands the hostname may be used in place of an IP address since the SSH client relies on the name resolving system of the operating system providing the translation of the hostname to its IP address equivalent. This resolution is normally achieved by use of DNS servers, local /etc/hosts file or other services involved in globally discovering names in the network. When you enter in ssh user@hostname your system queries the IP address that is mapped to hostname in these ways without connecting to the SSH daemon on the remote machine.

2. How secured is SSH?

- When properly set up, SSH is regarded to be extremely secure. It employs encrypted authentication credentials to guard against eavesdropping and man-in-the-middle attacks by encrypting all communication between the client and the server. SSH supports authentication mechanisms that need a high level of authentication, such as public-key authentication and multi-factor authentication. It employs powerful encryption techniques and safe key exchange, ensuring data security and integrity. The security relies on recommended practices such as blocking root login via SSH, using strong passwords or keys, keeping the server up to date, and carefully managing user access. When properly implemented, SSH is a tried-and-true method for providing secure remote services and administration.