Name: Santos, Emmanuelle Dave G.	**Date Performed: 08/08/2025**
Course/Section:CPE32S2	**Date Submitted:08/08/2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 2025-2026**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

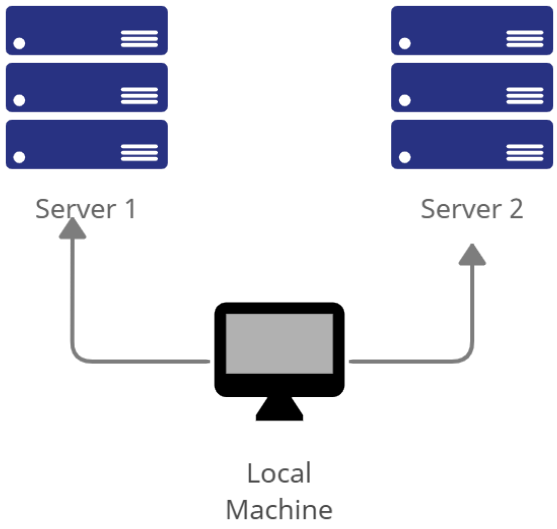
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

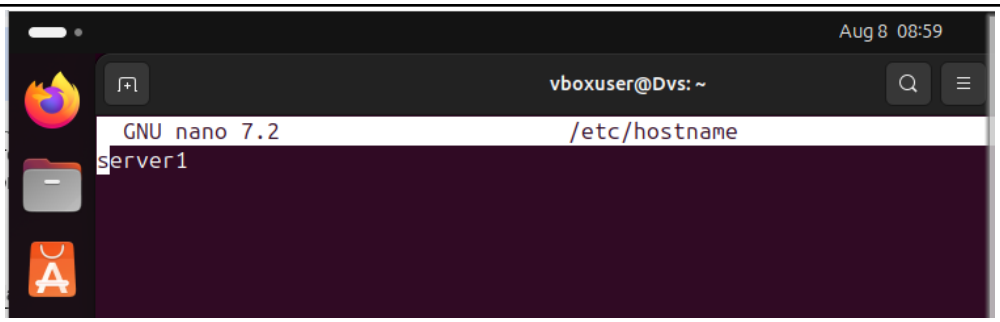
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



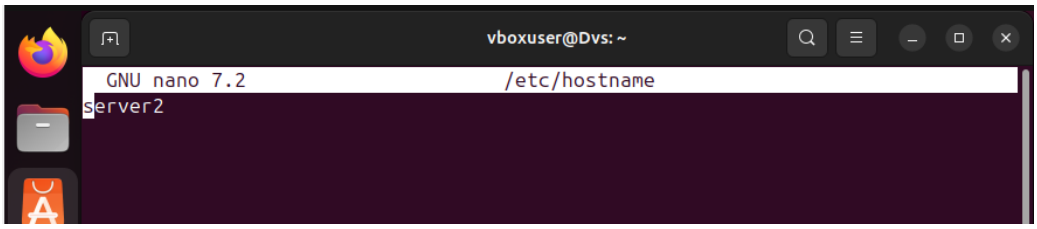
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

1. Change the hostname using the command *sudo nano /etc/hostname*

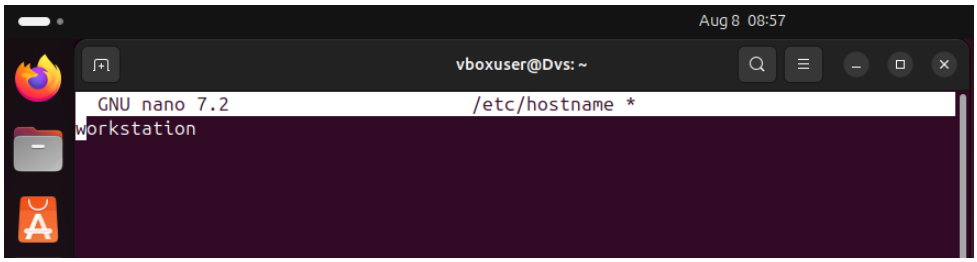
1.1 Use server1 for Server 1



1.2 Use server2 for Server 2



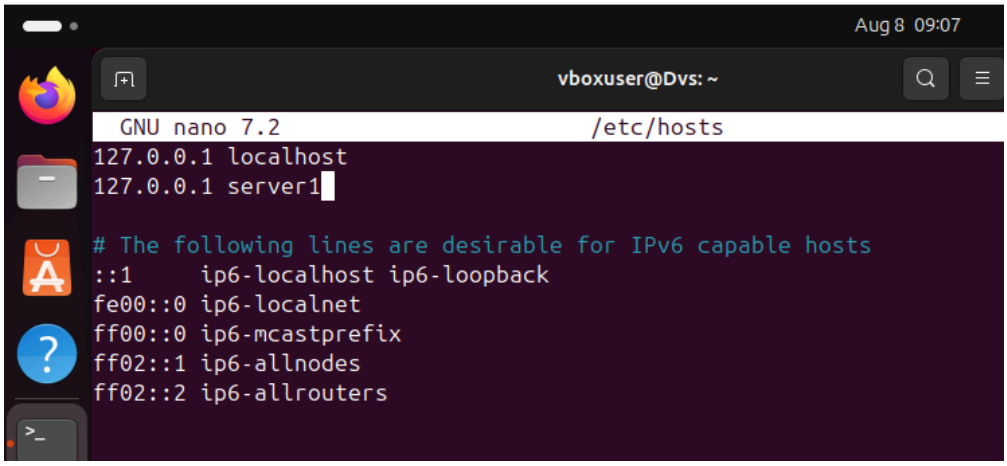
1.3 Use workstation for the Local Machine



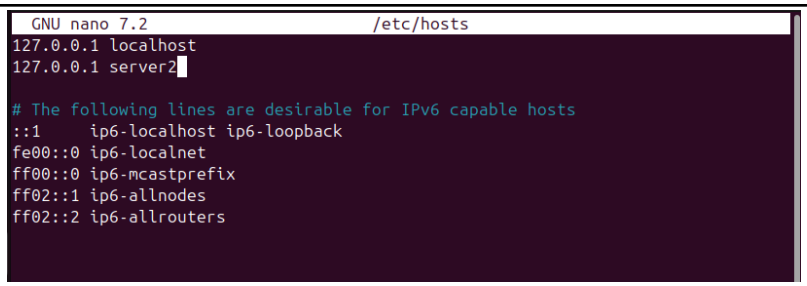
1.4

2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

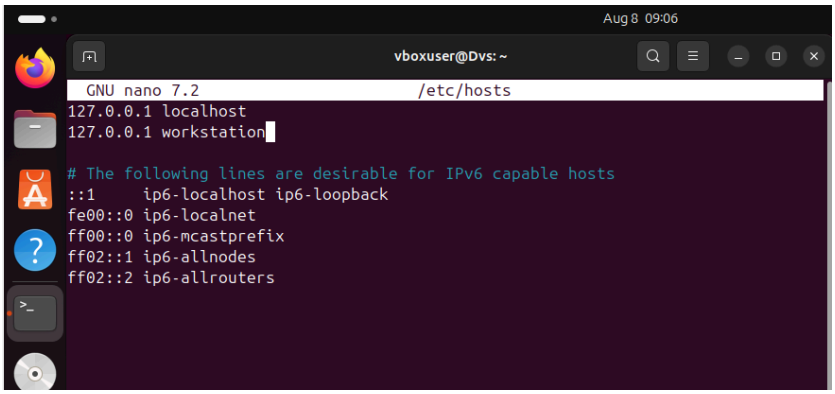
2.1 Type 127.0.0.1 server 1 for Server 1



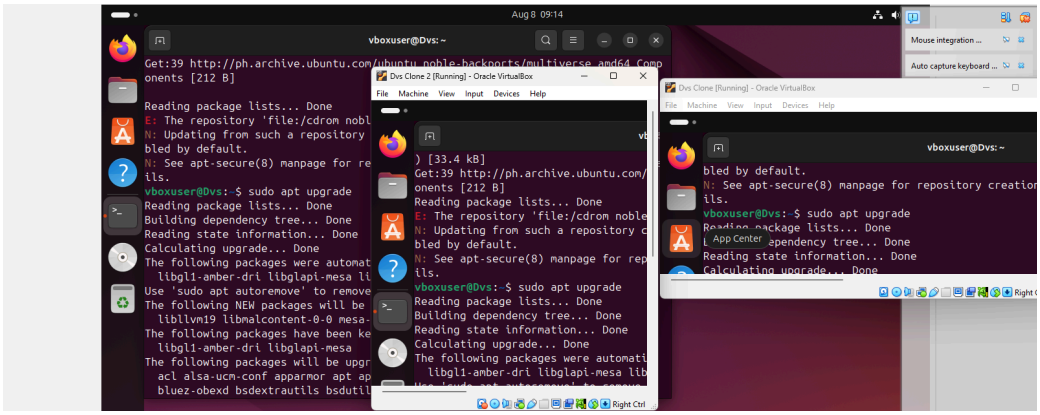
2.2 Type 127.0.0.1 server 2 for Server 2

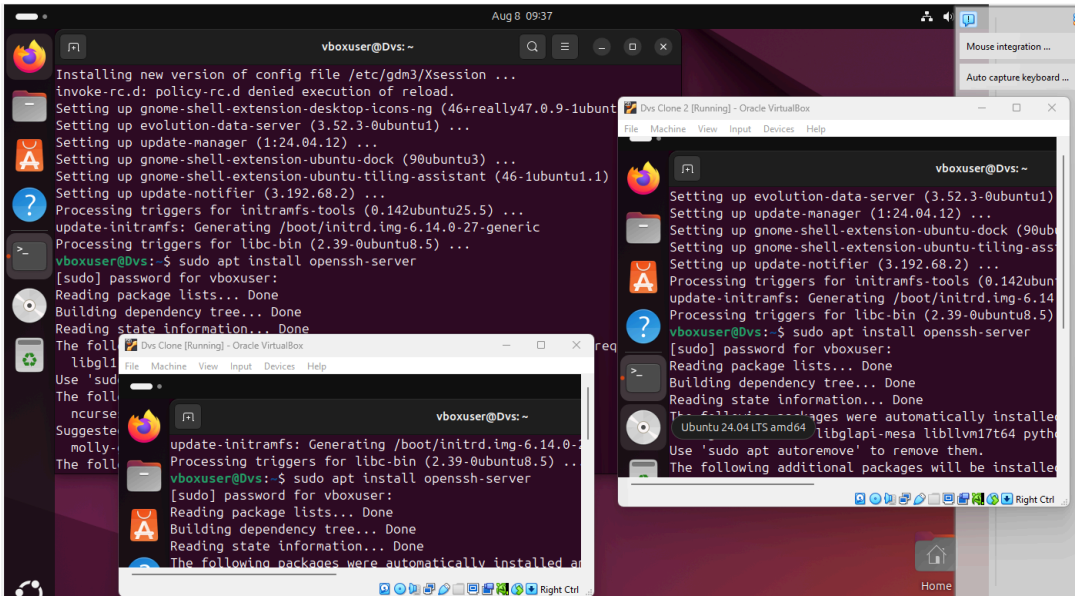


2.3 Type 127.0.0.1 workstation for the Local Machine



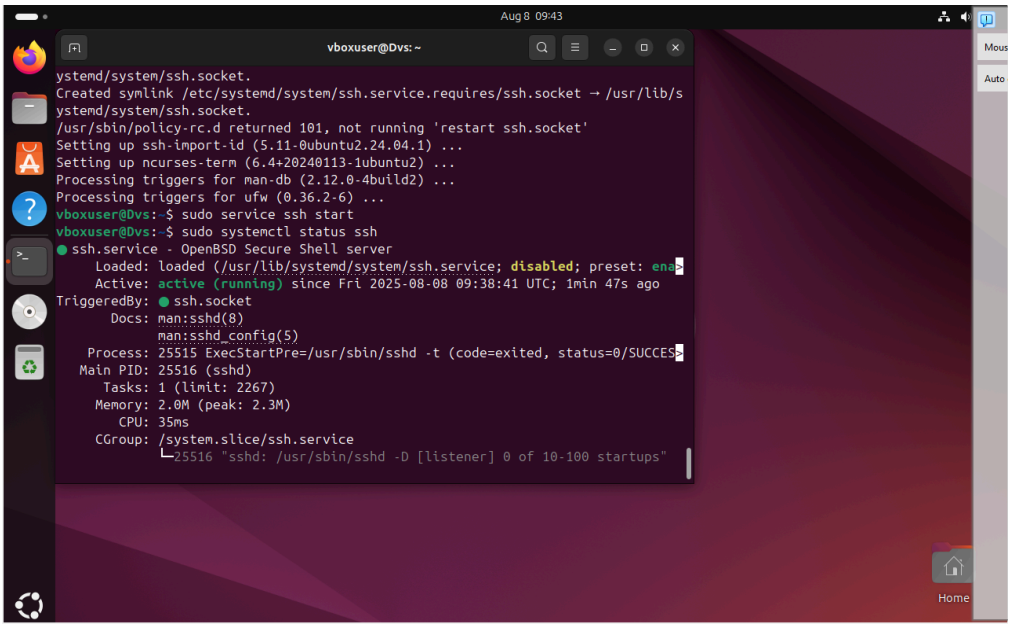
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.
2. 
3. Install the SSH server using the command *sudo apt install openssh-server*.

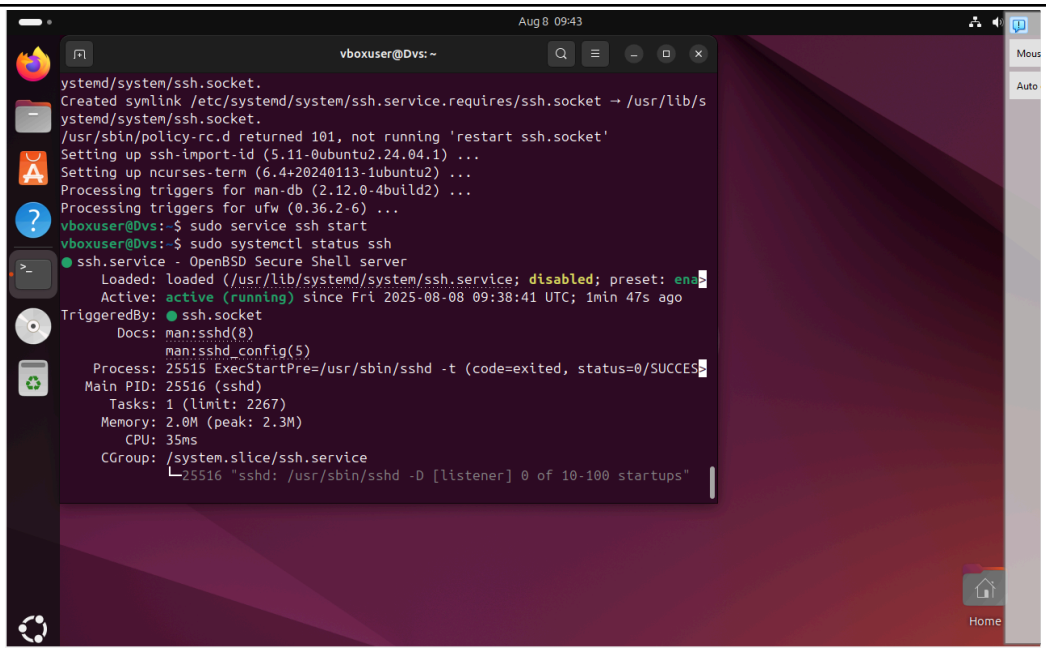


4. Verify if the SSH service has started by issuing the following commands:

   3.1 *sudo service ssh start*



   3.2 *sudo systemctl status ssh*



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

4.1 *sudo ufw allow ssh*

```
vboxuser@Dvs:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
vboxuser@Dvs:~$
```

4.2 *sudo ufw enable*

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

4.3 *sudo ufw status*

```
vboxuser@Dvs:~$ 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.**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

```
vboxuser@workstation:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=2.59 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.11 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=1.08 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.942 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.538 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.758 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.529 ms
```

2.2

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

```
vboxuser@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=6.70 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.899 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.520 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.09 ms
^C
```

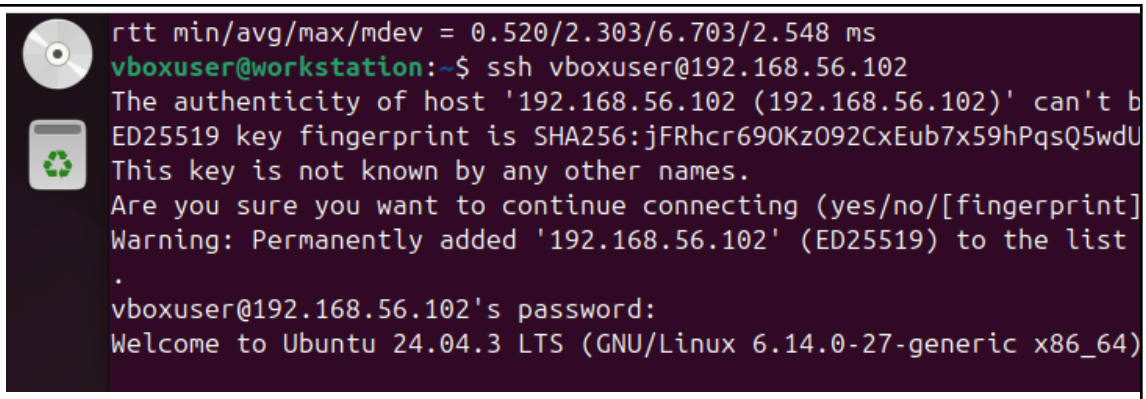
2.4 Connectivity test for Server 1 to Server 2: ☐ Successful ☐ Not Successful

```
vboxuser@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=1.64 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.871 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.823 ms
^C
--- 192.168.56.103 ping statistics ---
```

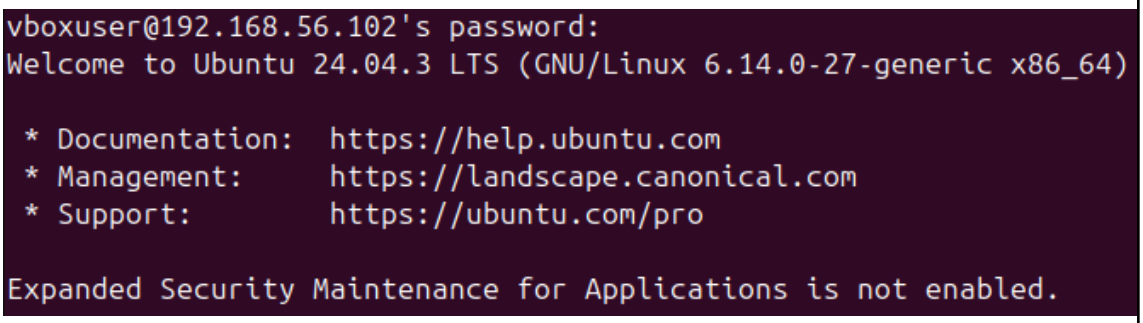
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

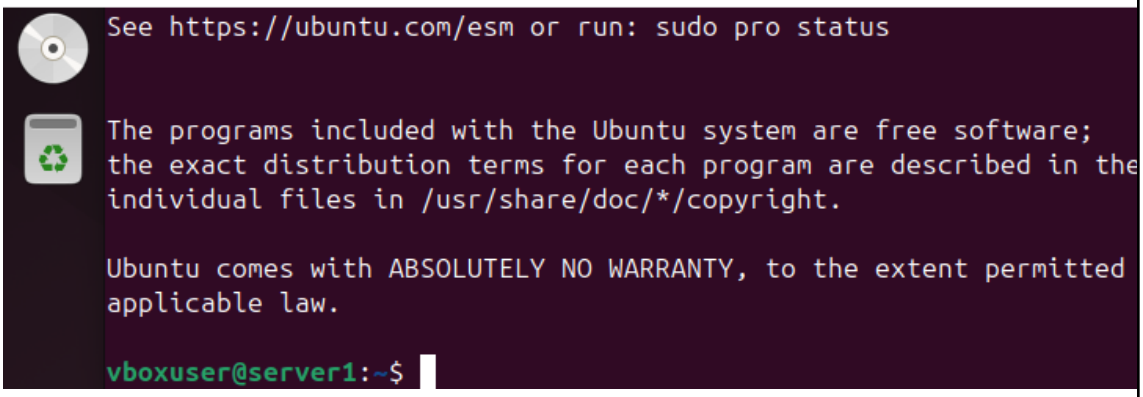
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*



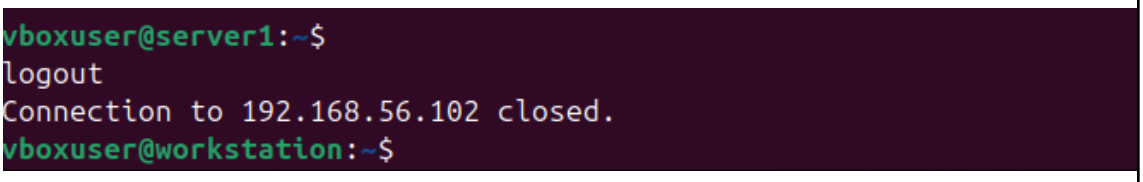
1.2 Enter the password for server 1 when prompted



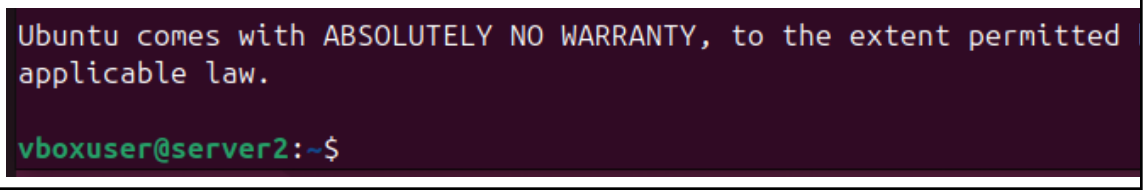
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*



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



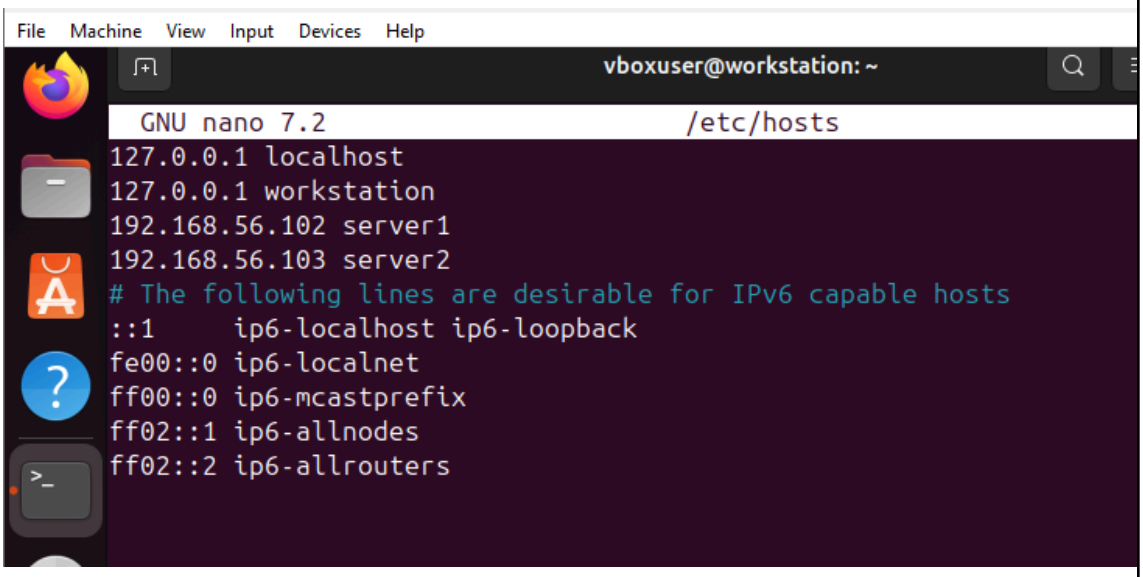
3. Do the same for Server 2.



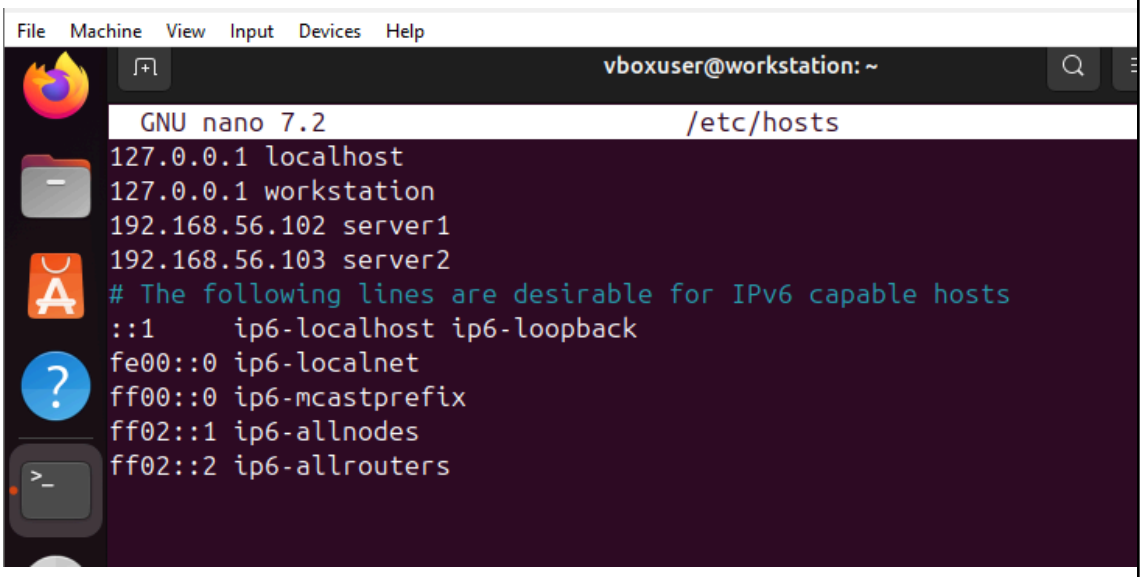
```
vboxuser@server2:~$
logout
Connection to 192.168.56.103 closed.
vboxuser@workstation:~$
```

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

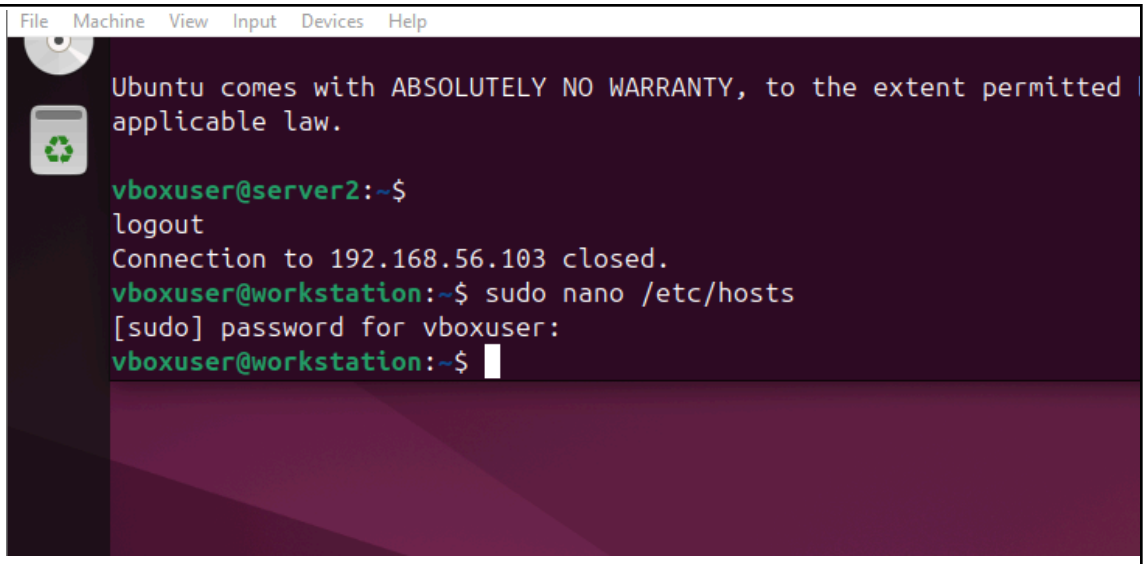
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)



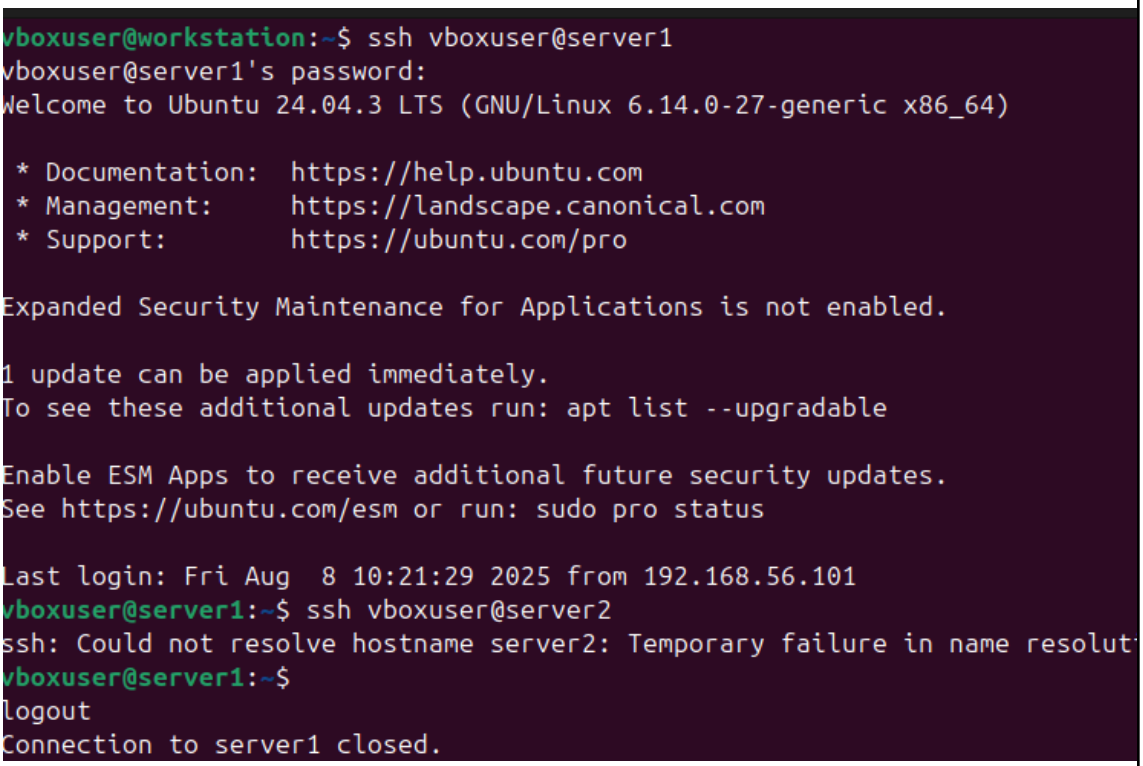
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

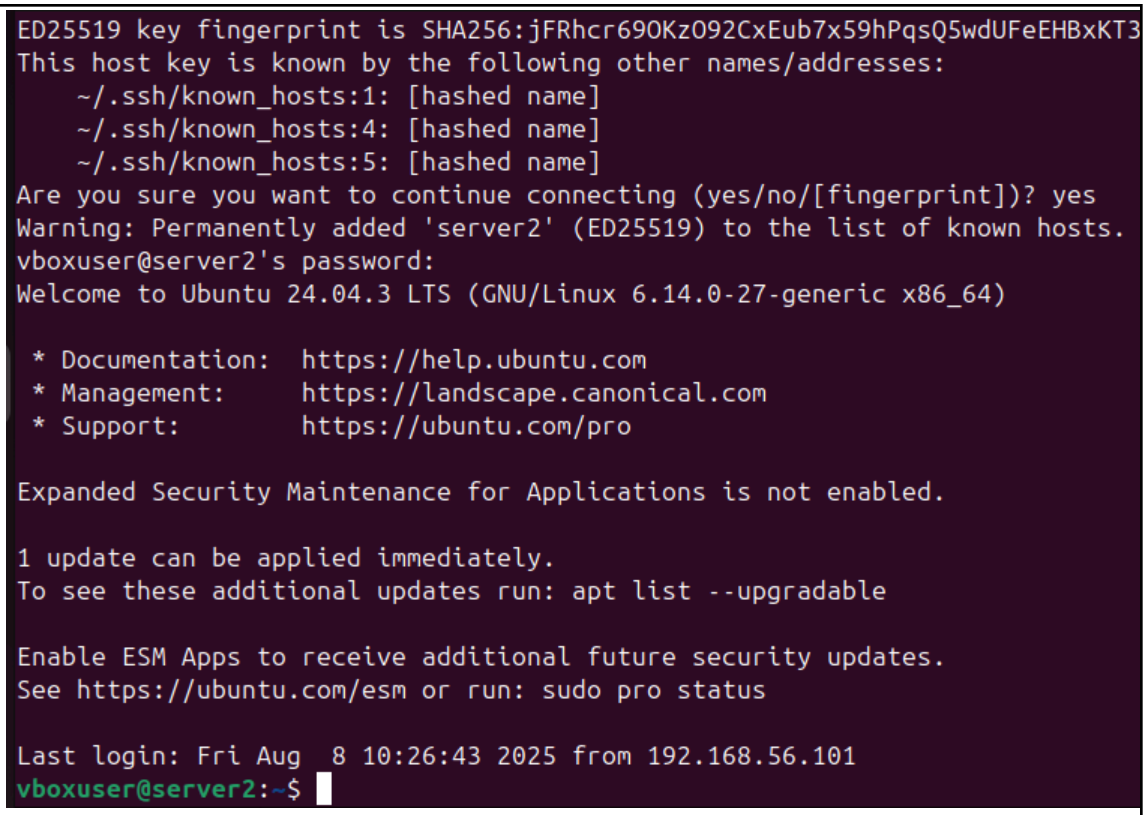


4.3 Save the file and exit.



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?
   because the IP address is local to the the servers in can be used instead of names
2. How secured is SSH?

   SSH is secured because it always required a password if you create a user ssh and cannot not easily bybass