Name: Edgardo Kenneth D. Cordero	**Date Performed: 8/16/2022**
Course/Section: CPE 232 - CPE31S22	**Date Submitted: 8/23/2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 1ST SEMESTER**

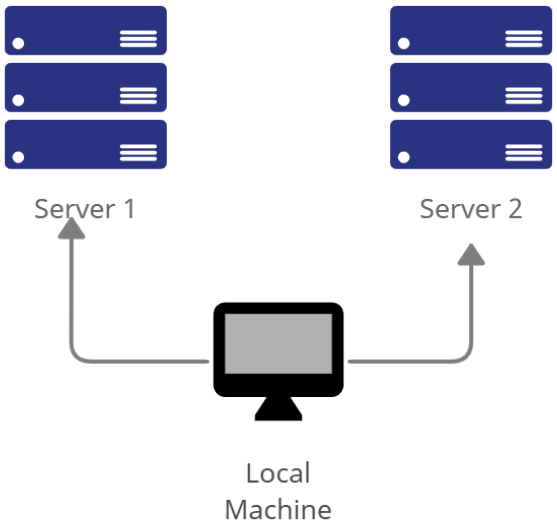
Activity 1: Configure Network using Virtual Machines

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:

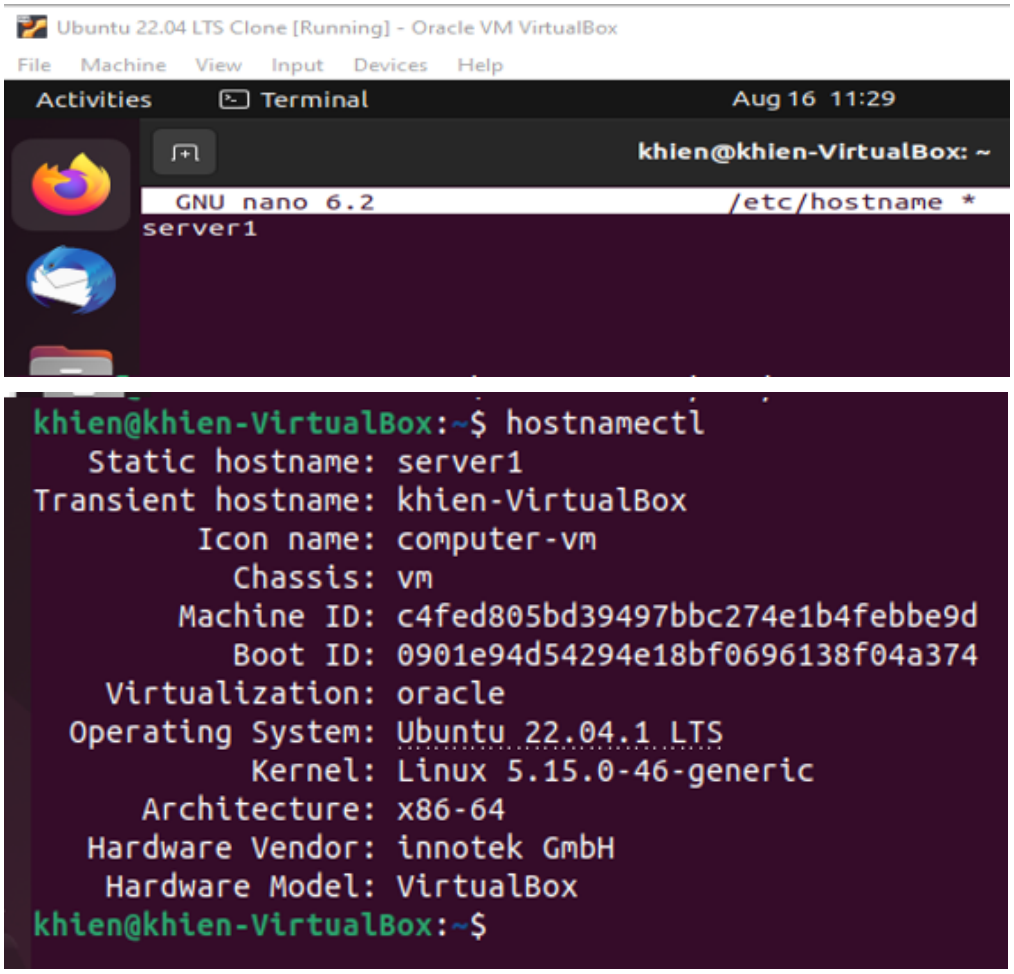
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



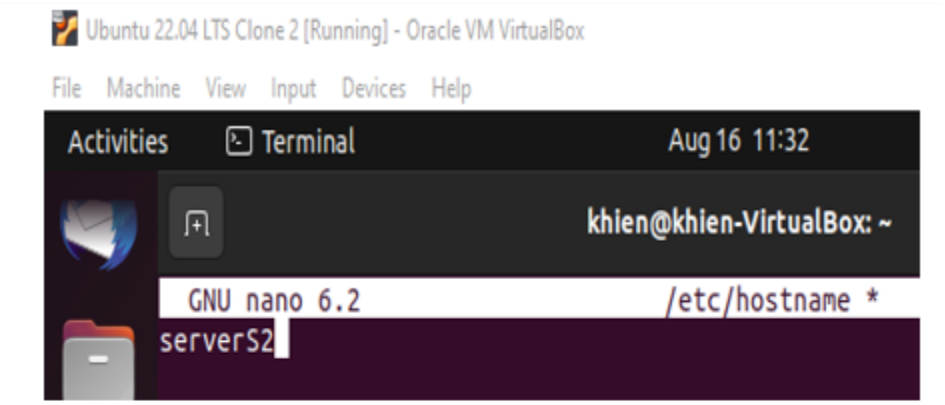
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

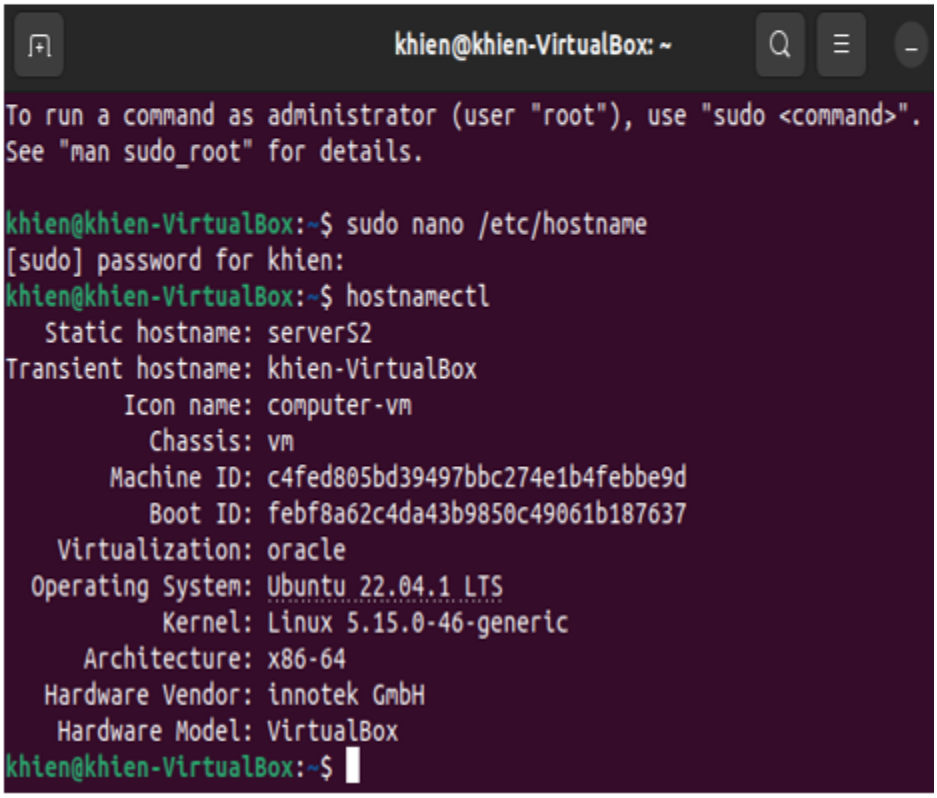
1. Change the hostname using the command *sudo nano /etc/hostname*

1.1 Use server1 for Server 1

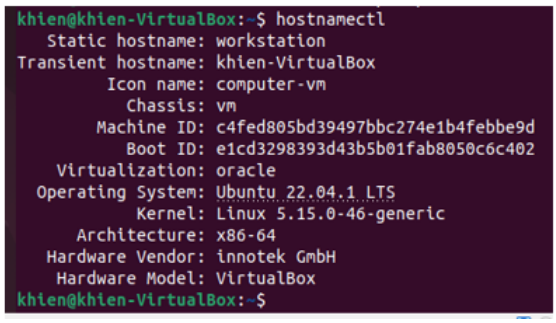


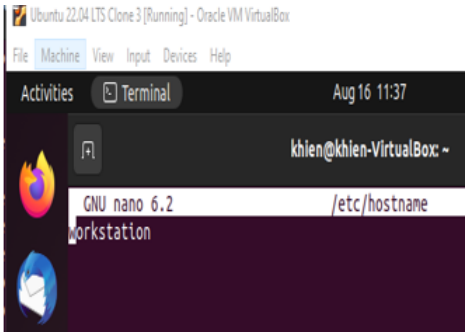
1.2 Use server2 for Server 2





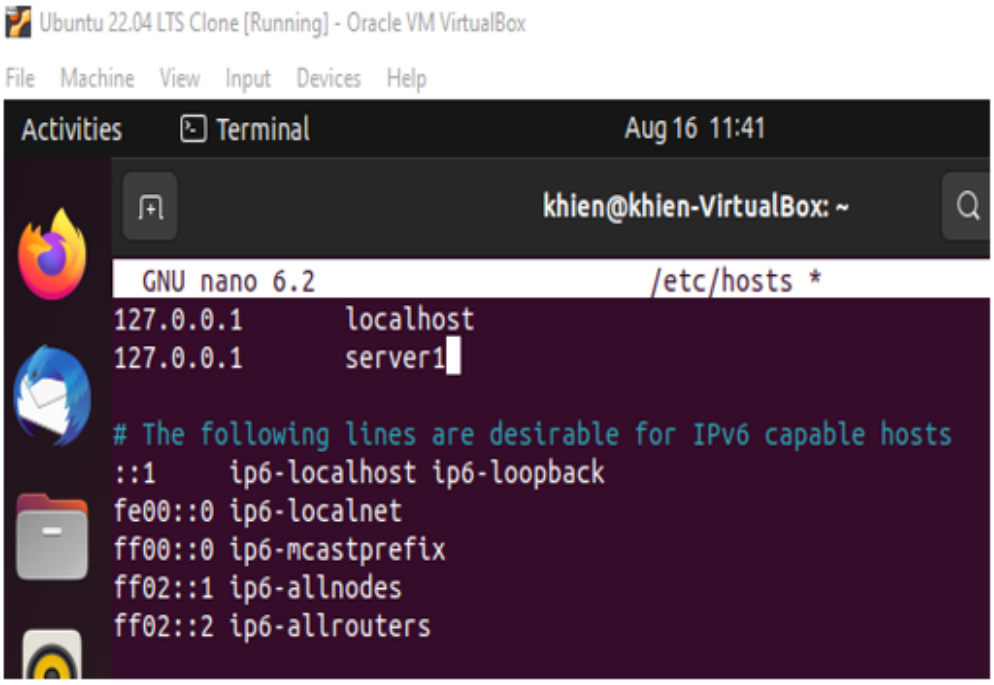
1.3 Use workstation for the Local Machine



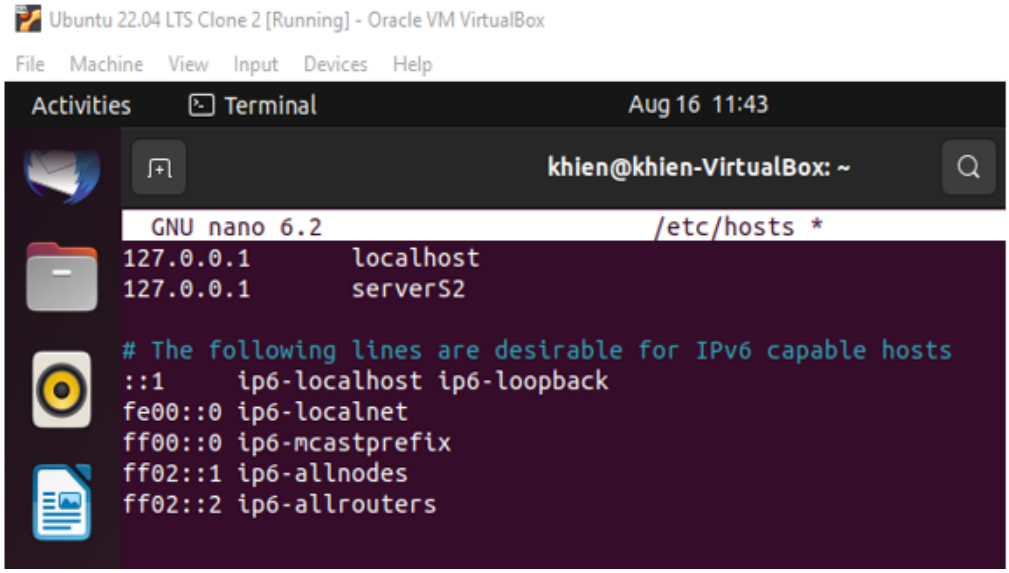


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

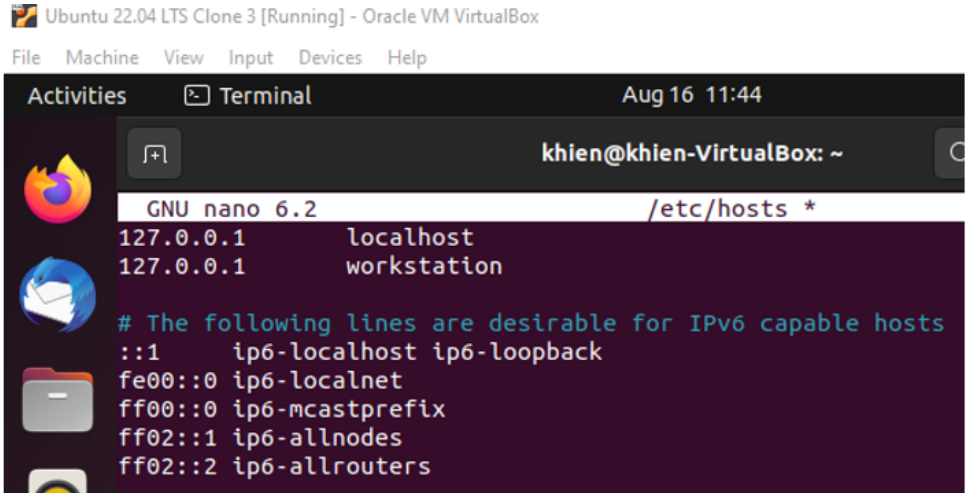
2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2

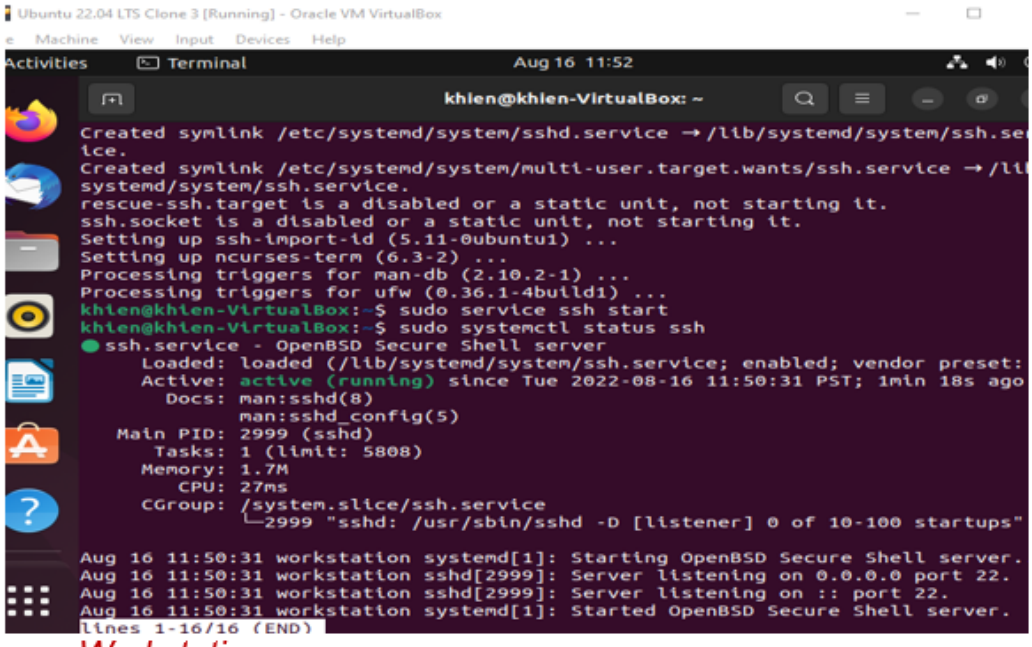


2.3 Type 127.0.0.1 workstation for the Local Machine

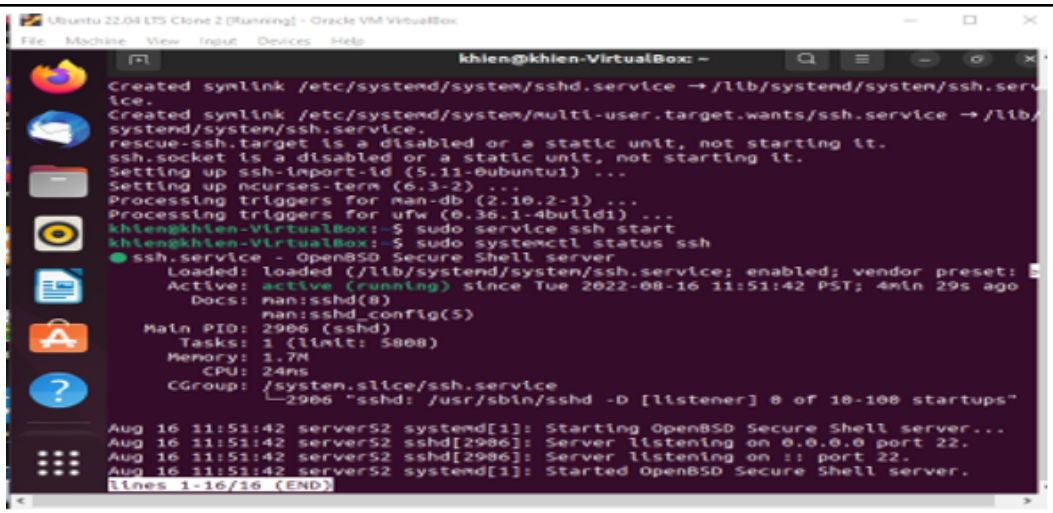


Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

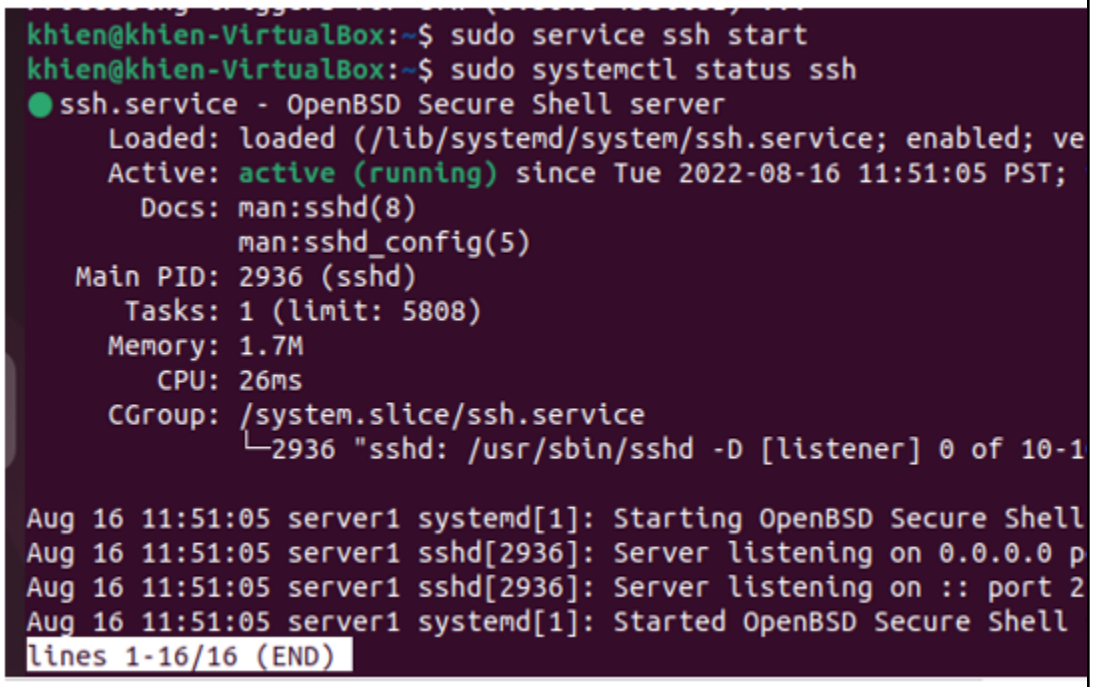
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.
2. Install the SSH server using the command *sudo apt install openssh-server*.
3. Verify if the SSH service has started by issuing the following commands:
   3.1 sudo service ssh start
   3.2 sudo systemctl status ssh



Workstation



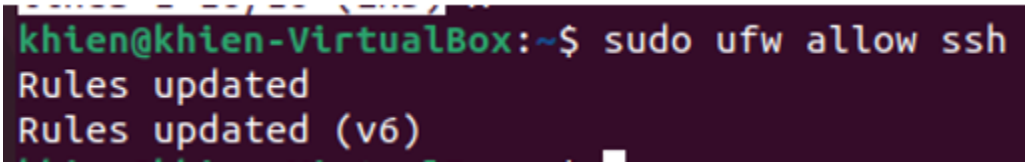
serverS2



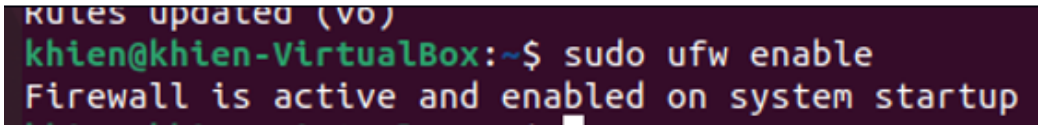
Server1

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

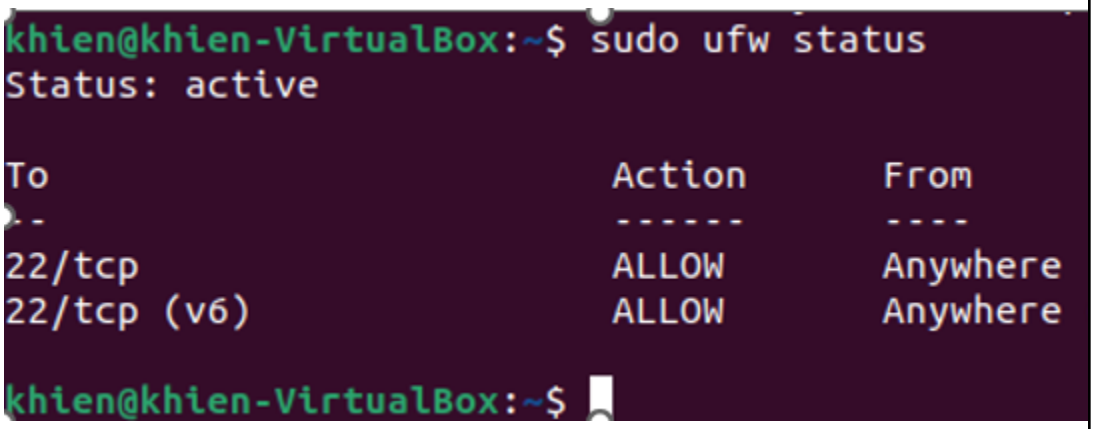
4.1 sudo ufw allow ssh



4.2 sudo ufw enable



4.3 sudo ufw status



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: 192.168.56.101
   1.3 Server 3 IP address: 192.168.56.108
2. Make sure that they can ping each other.

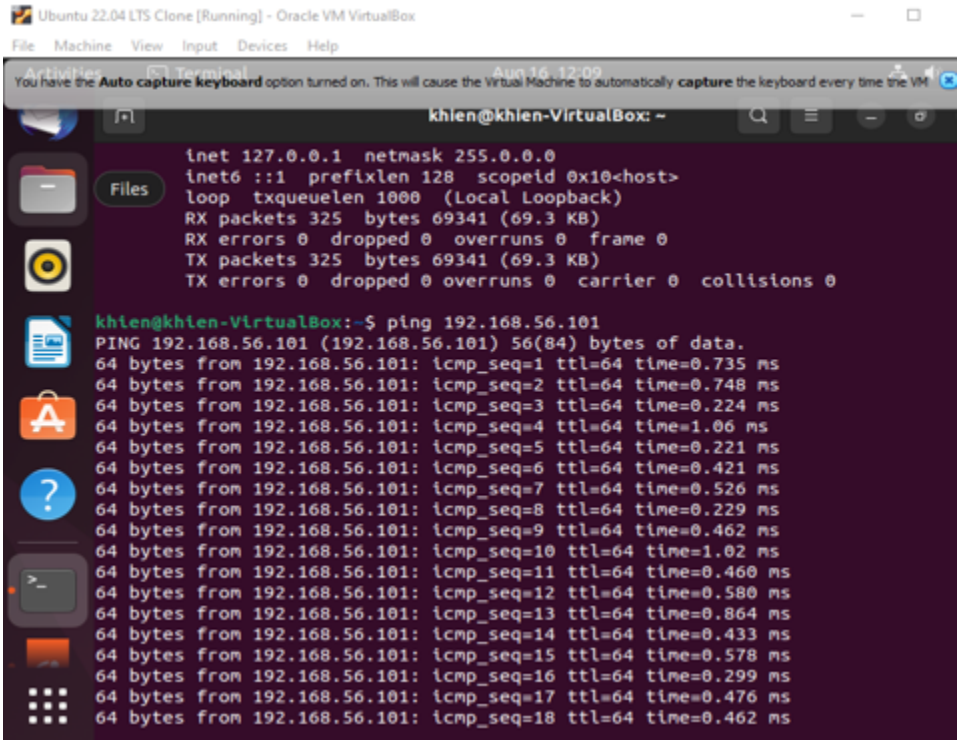
2.1 Connectivity test for Local Machine 1 to Server 1: ✅ Successful

```
khien@serverS2:~$ 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=0.409 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.435 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.274 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.426 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.265 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.289 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.393 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.337 ms
64 bytes from 192.168.56.102: icmp_seq=10 ttl=64 time=0.402 ms
64 bytes from 192.168.56.102: icmp_seq=11 ttl=64 time=0.396 ms
64 bytes from 192.168.56.102: icmp_seq=12 ttl=64 time=0.392 ms
64 bytes from 192.168.56.102: icmp_seq=13 ttl=64 time=0.240 ms
64 bytes from 192.168.56.102: icmp_seq=14 ttl=64 time=0.700 ms
64 bytes from 192.168.56.102: icmp_seq=15 ttl=64 time=0.612 ms
64 bytes from 192.168.56.102: icmp_seq=16 ttl=64 time=0.372 ms
64 bytes from 192.168.56.102: icmp_seq=17 ttl=64 time=0.365 ms
64 bytes from 192.168.56.102: icmp_seq=18 ttl=64 time=0.347 ms
^C
--- 192.168.56.102 ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 17408ms
rtt min/avg/max/mdev = 0.240/0.386/0.700/0.111 ms
```

2.2 Connectivity test for Local Machine 1 to Server 2: ✅ Successful

```
khien@serverS2:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.454 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.485 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.380 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.497 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.424 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.374 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.527 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.245 ms
64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.366 ms
64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.457 ms
64 bytes from 192.168.56.101: icmp_seq=11 ttl=64 time=0.255 ms
64 bytes from 192.168.56.101: icmp_seq=12 ttl=64 time=0.387 ms
64 bytes from 192.168.56.101: icmp_seq=13 ttl=64 time=0.428 ms
64 bytes from 192.168.56.101: icmp_seq=14 ttl=64 time=0.311 ms
^C
--- 192.168.56.101 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13314ms
rtt min/avg/max/mdev = 0.245/0.399/0.527/0.082 ms
khien@serverS2:~$
```

2.3 Connectivity test for Server 1 to Server 2: ✅ Successful



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

Server 1 IP address: 192.168.56.102

Server 2 IP address: 192.168.56.101

Server 3 IP address: 192.168.56.108

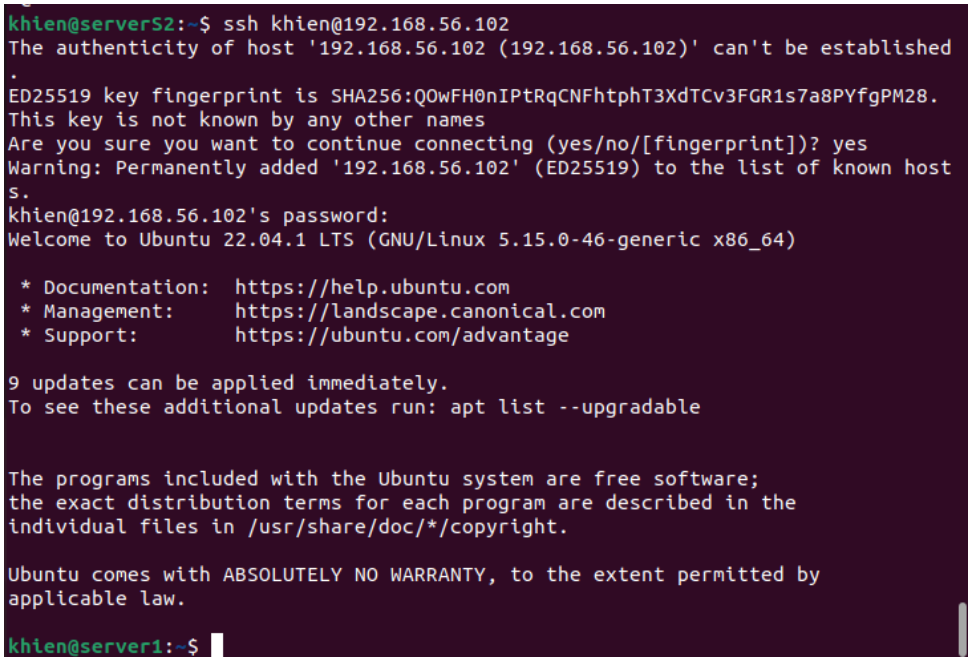
1. On the Local Machine, issue the following commands:

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

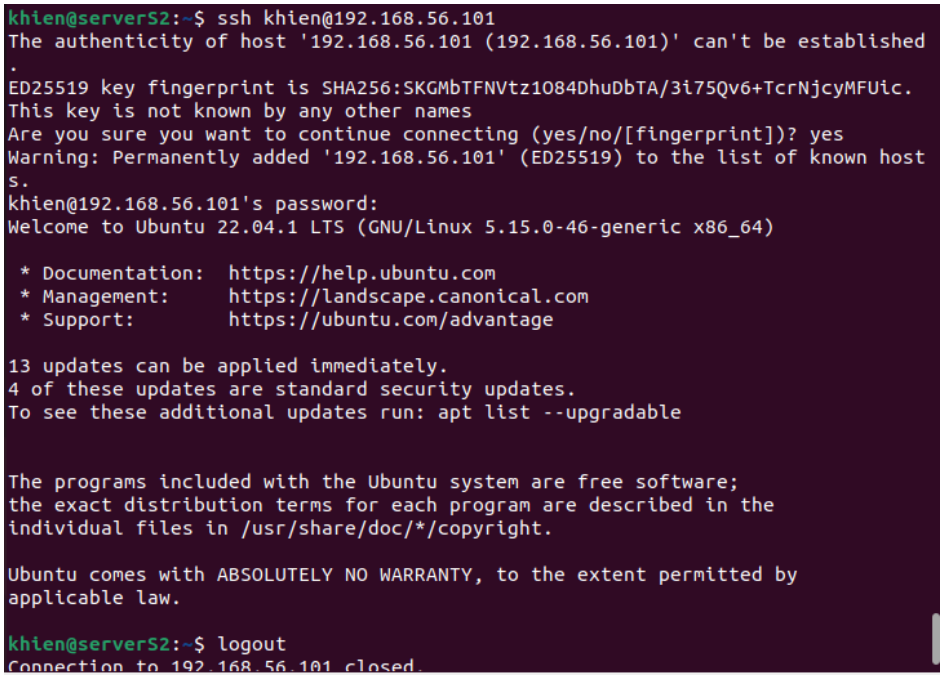
1.2 Enter the password for server 1 when prompted

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.*



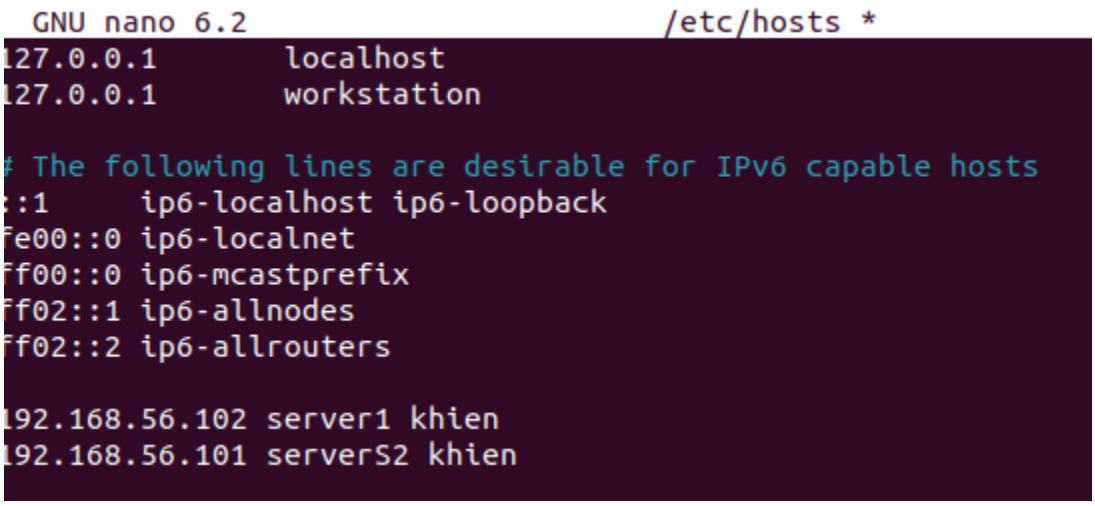
3. Do the same for Server 2.



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

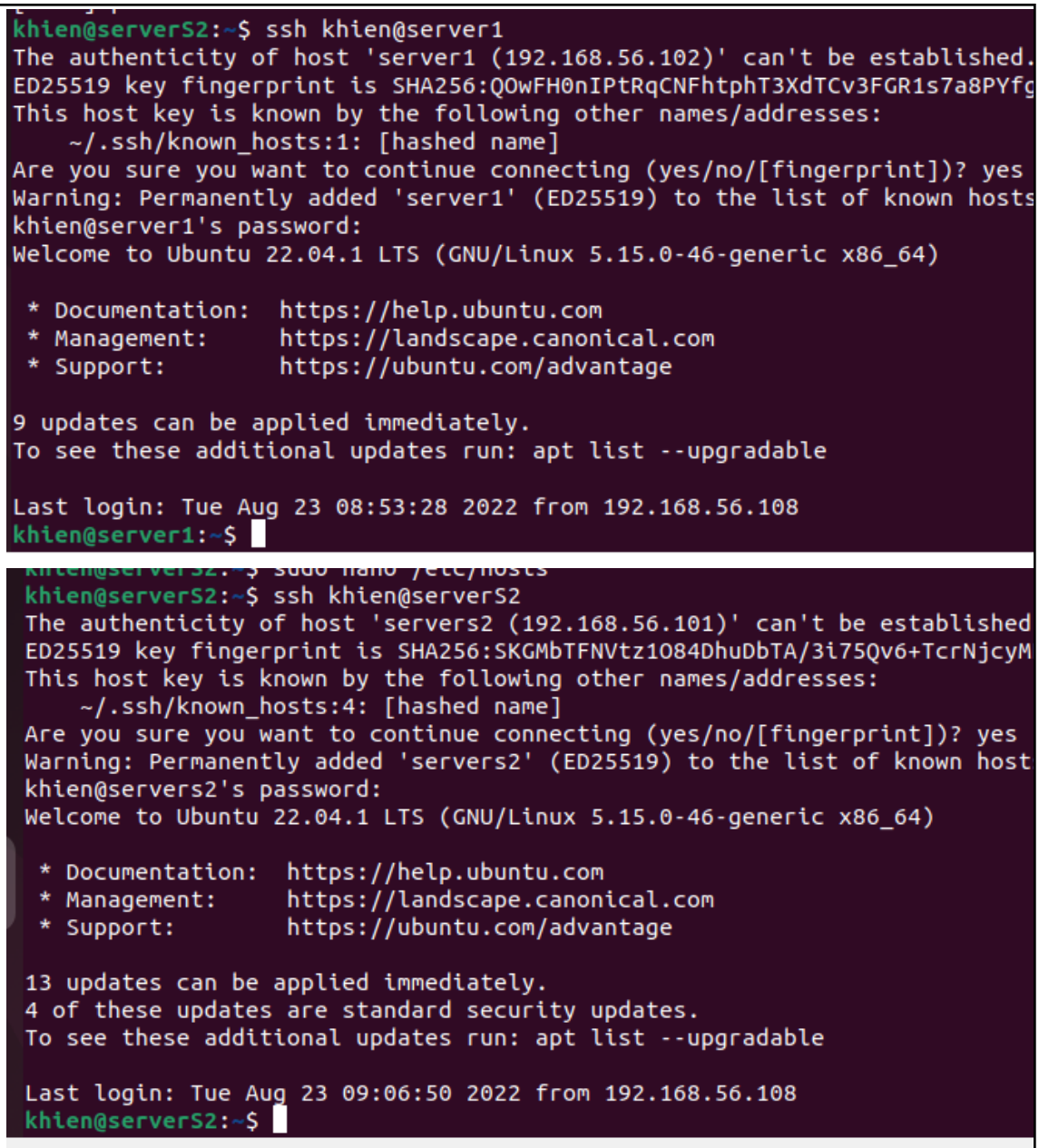
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

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?
   To be able to use the hostname instead of IP address in SSH commands you need to use the local machine and edit the the hosts by using the commands of sudo nano /etc/hosts and put the ip address of the server and the server name and the user and the format for that is (ip address) (server name) (user name)
2. How secured is SSH?
   The security of SSH is one of the highly secured of the standard security precautions. It is the most significant application because it is a remote login and

command line execution. It is also considered as a secure alternative to unsecured remote shell protocols that can utilize the client server paradigm in which it has a secure channel.