Name: JOZETTE FERMIN	**Date Performed: august 14, 2023**
Course/Section: august 15, 2023	**Date Submitted: august 15, 2023**
Instructor: Engr. Jonathan Taylar	**Semester and SY: 1st sem 2023 - 2024**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

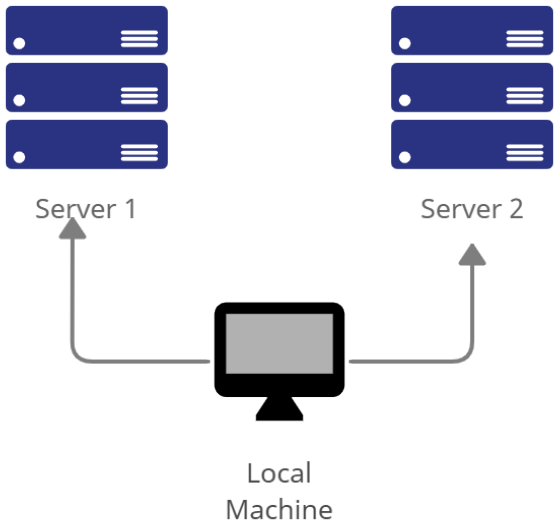
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

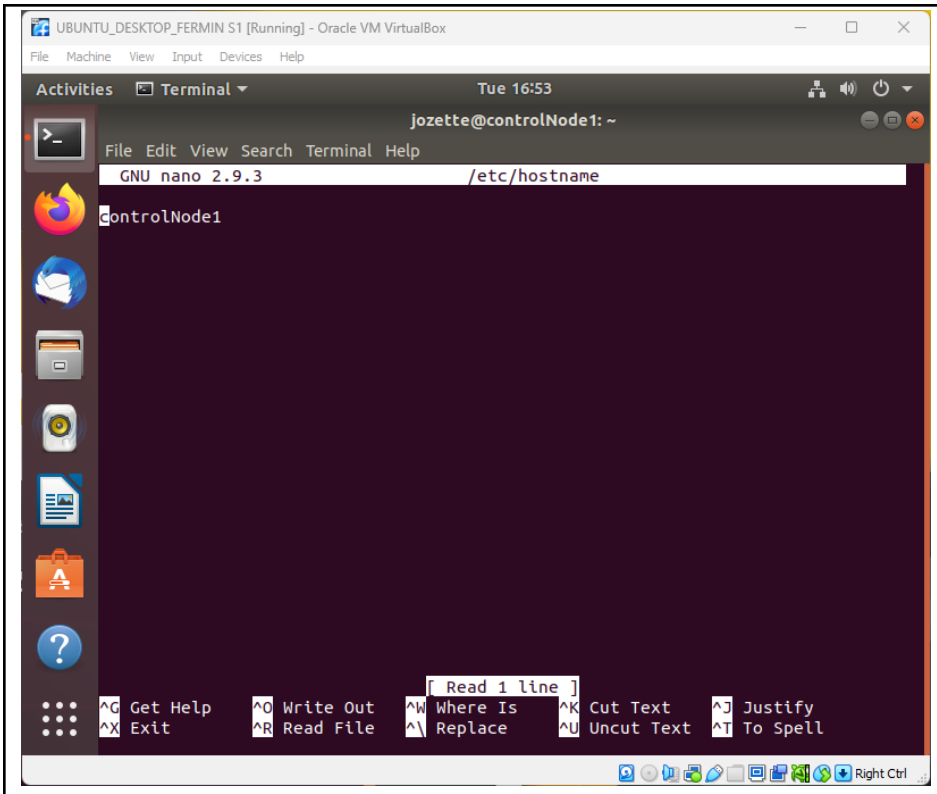
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



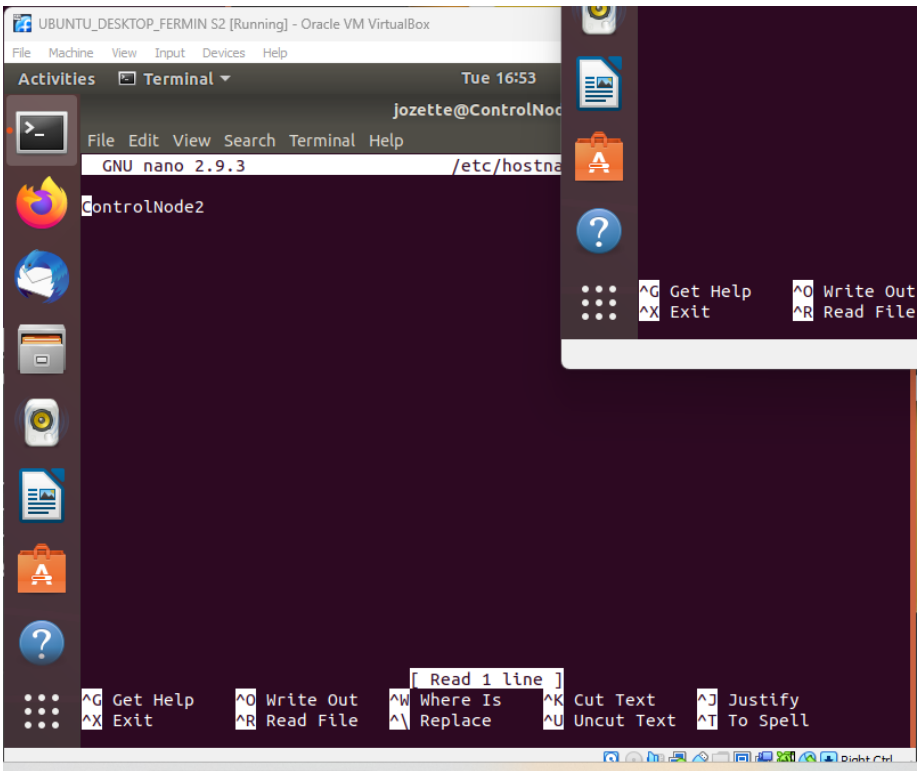
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

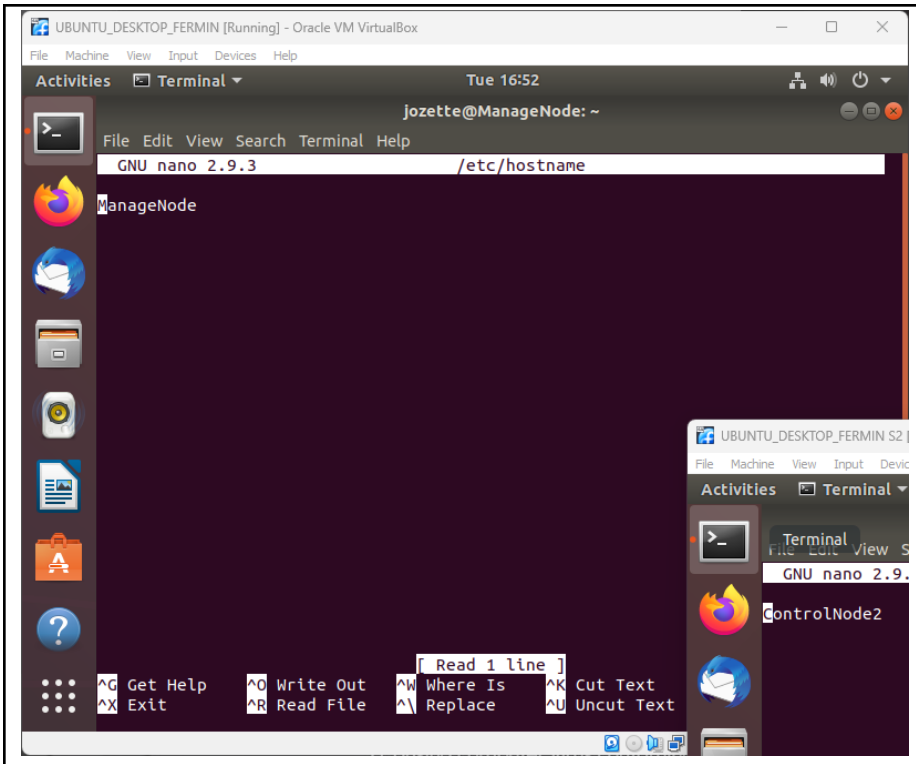
1.1 Use server1 for Server 1



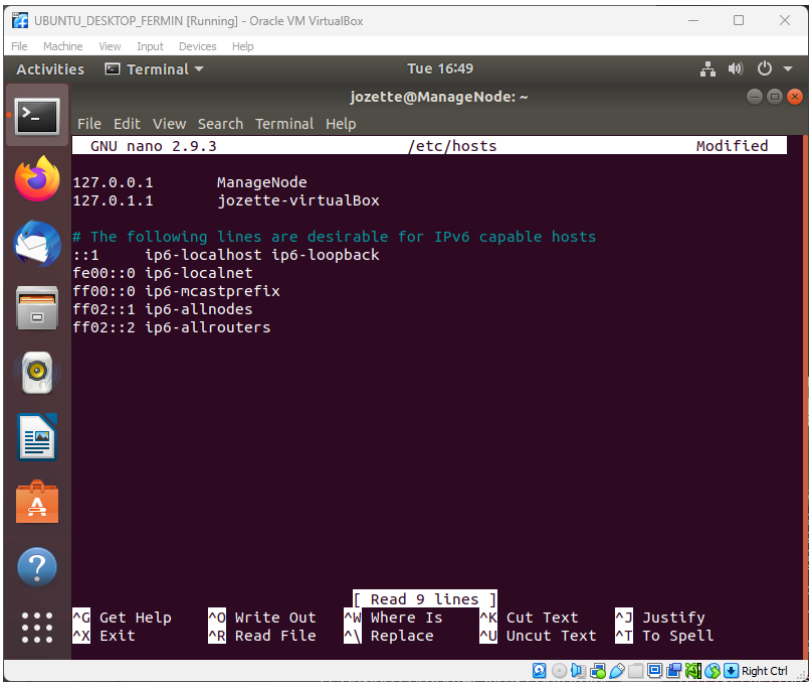
1.2 Use server2 for Server 2



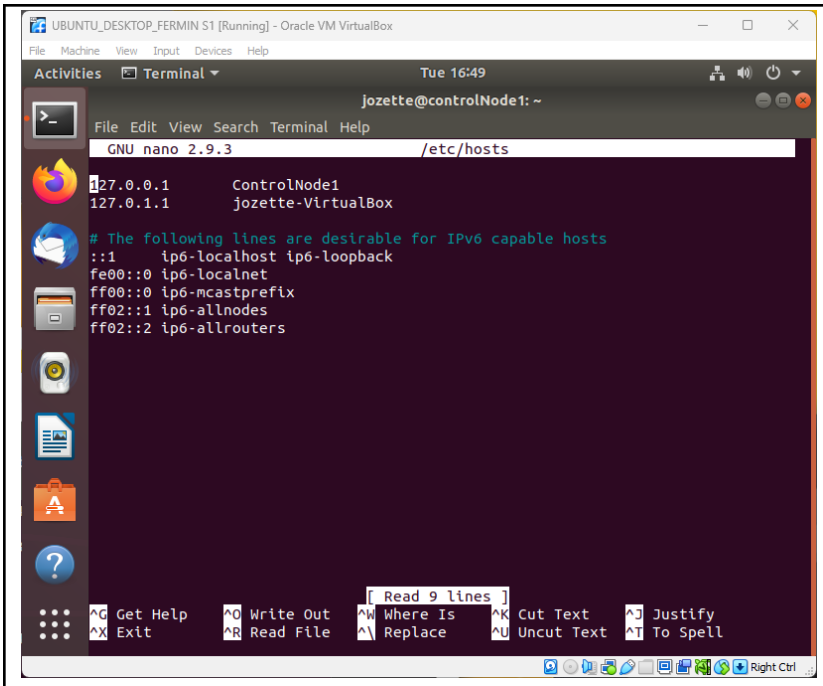
1.3 Use workstation for the Local Machine



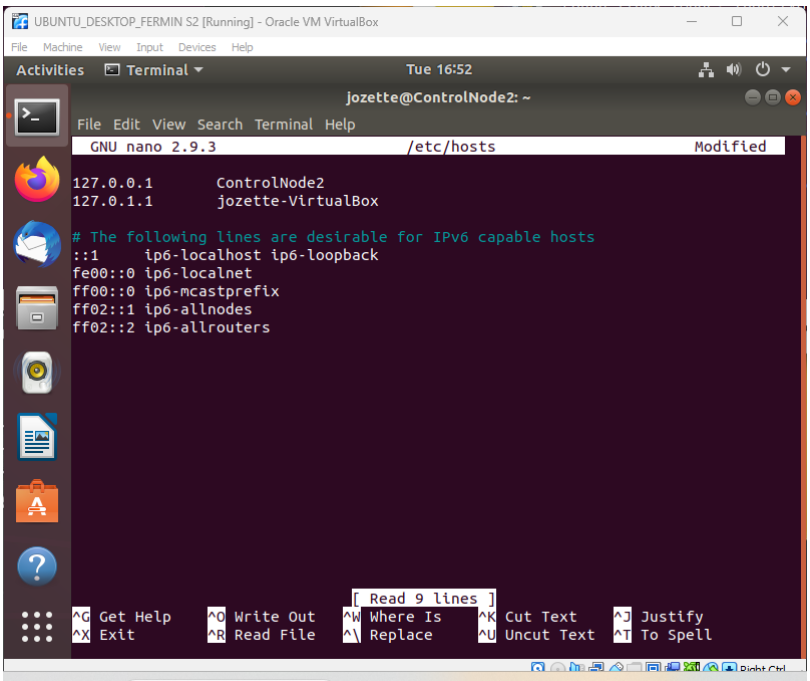
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1



   2.2 Type 127.0.0.1 server 2 for Server 2

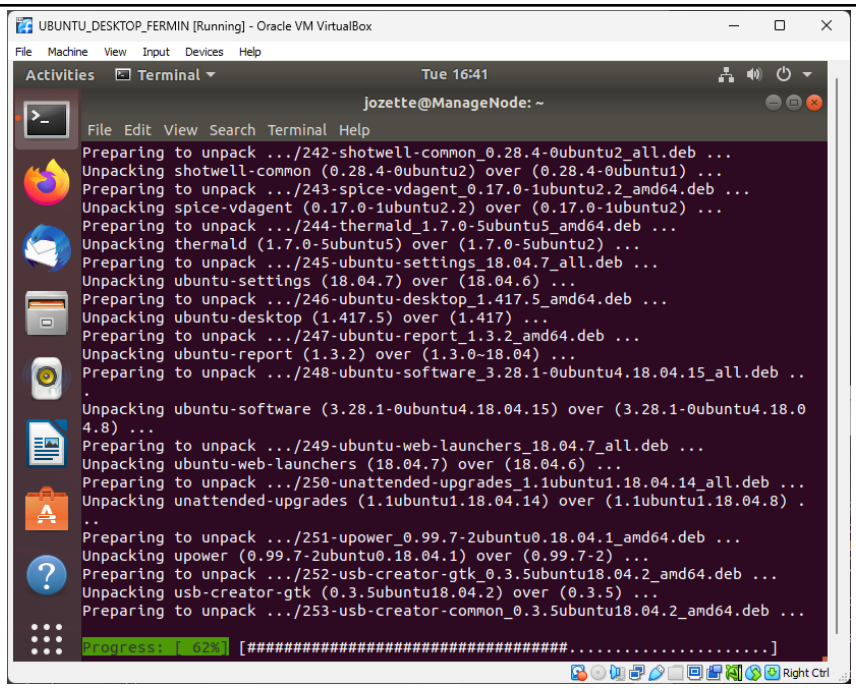


2.3 Type 127.0.0.1 workstation for the Local Machine

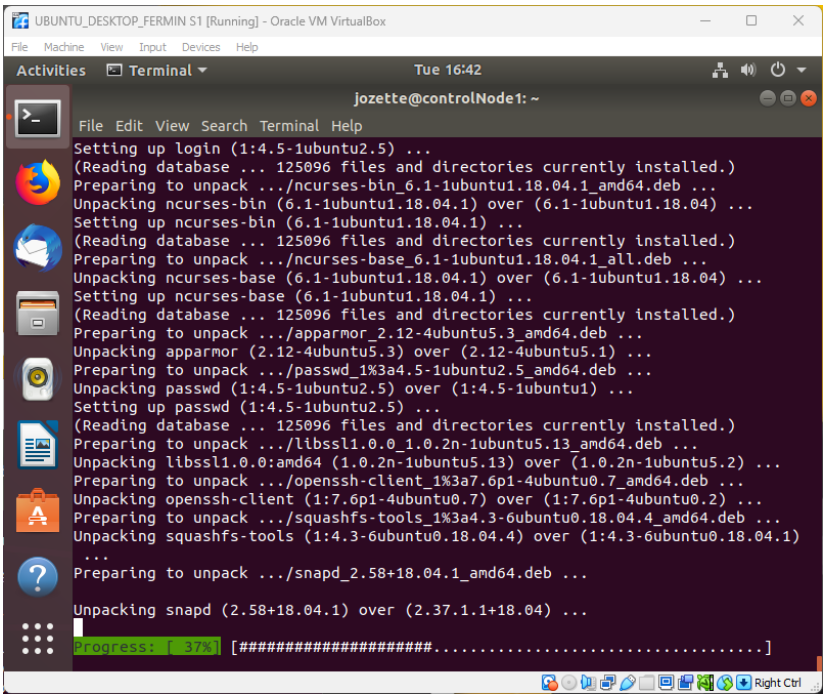


Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

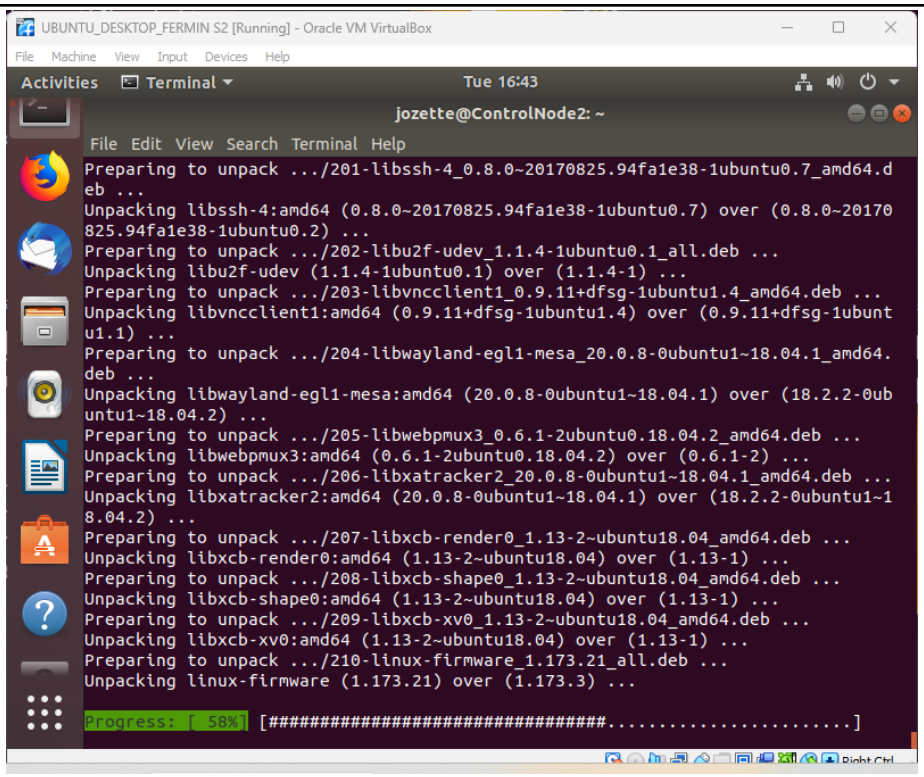
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.



ManageNode

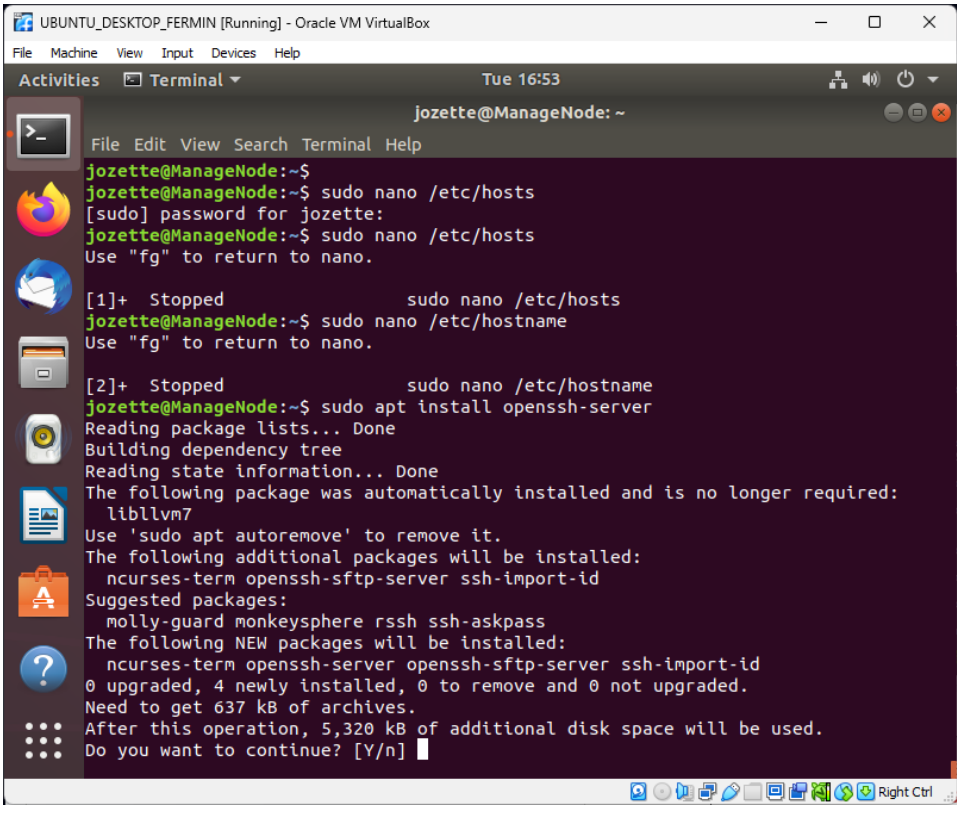


ControlNode1

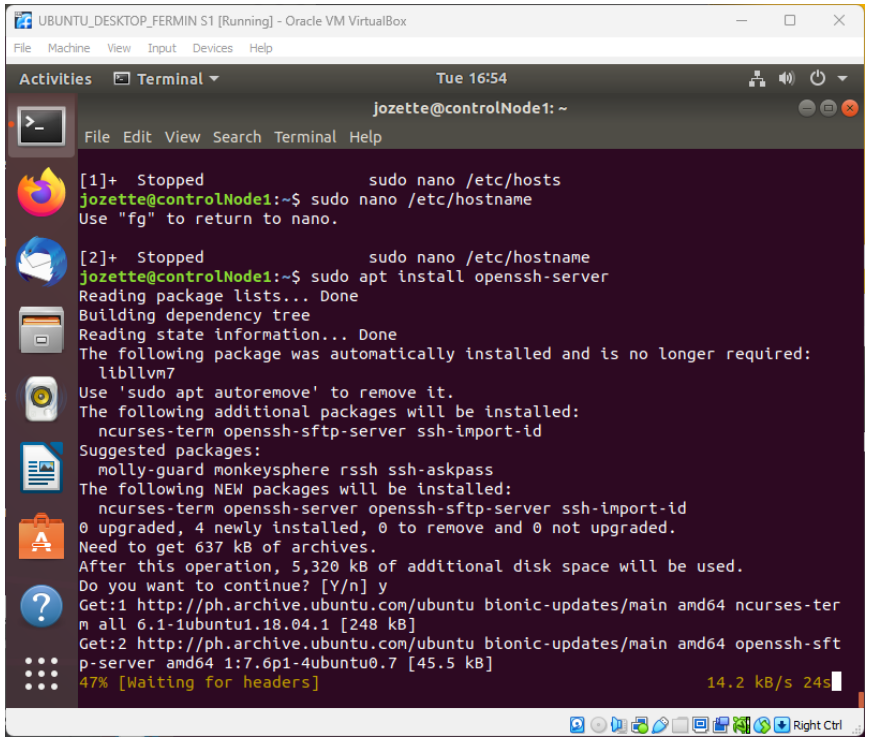


ControlNode2

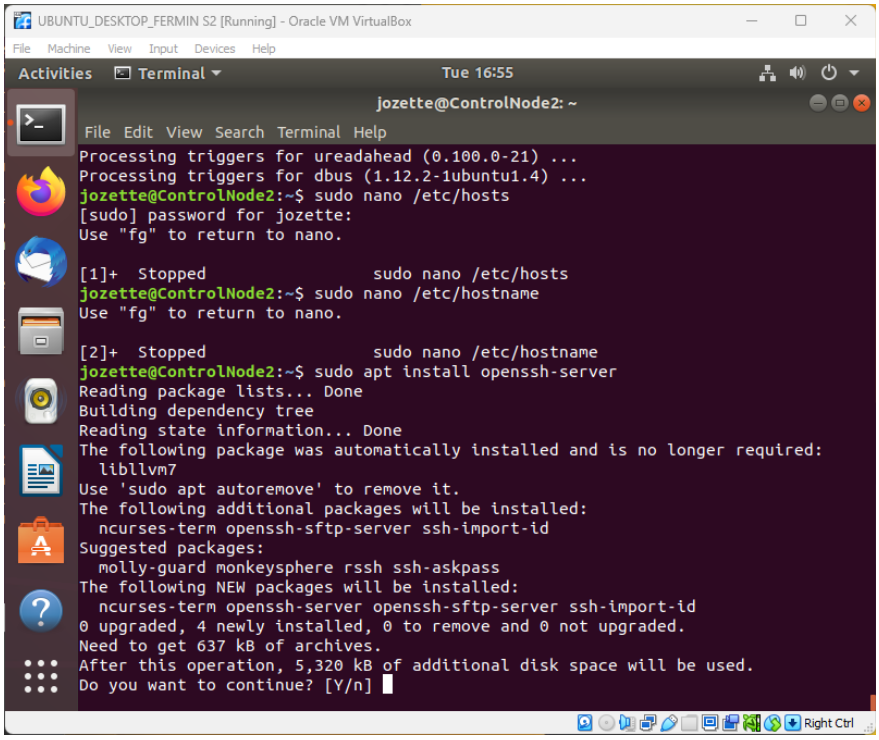
2. Install the SSH server using the command *sudo apt install openssh-server*.



ManageNode



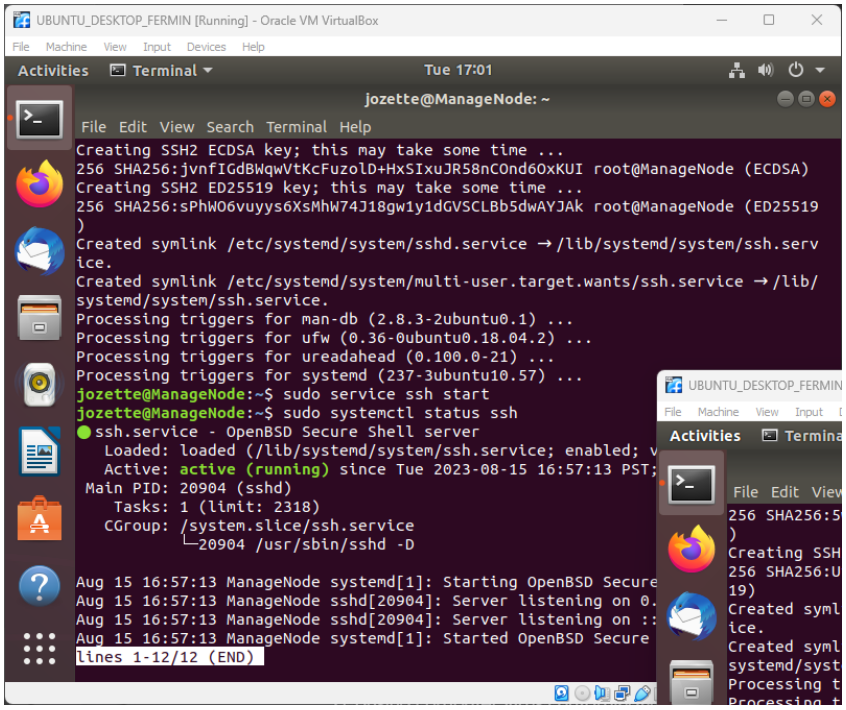
ControlNode1



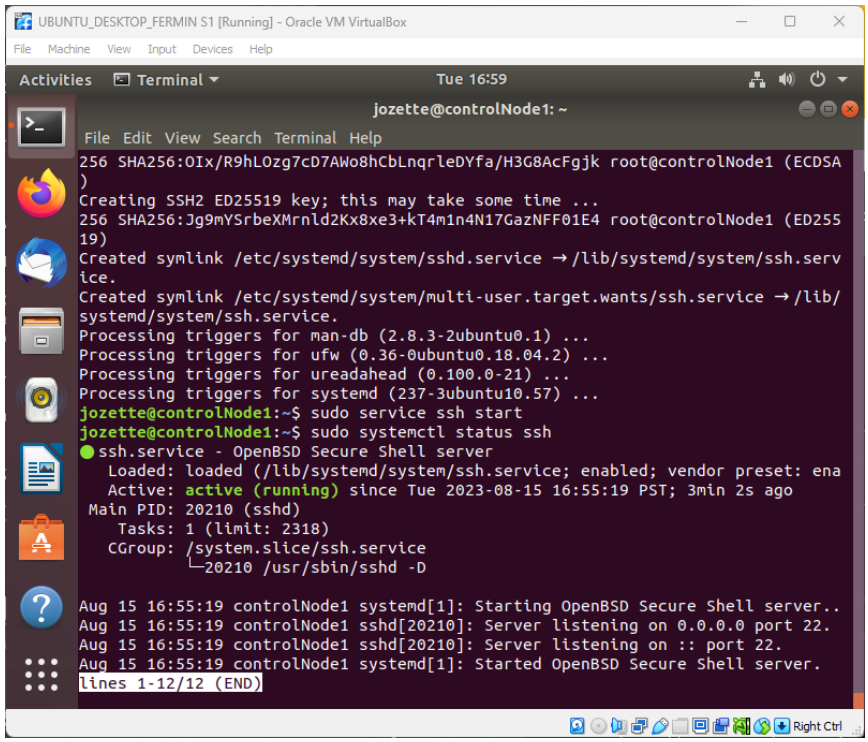
ControlNode2

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

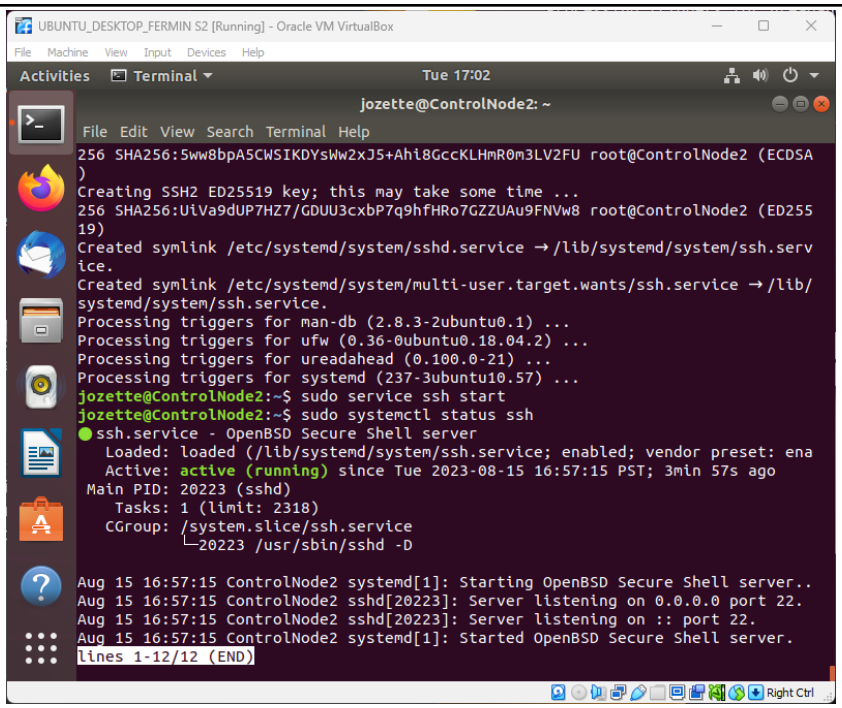
3.2 *sudo systemctl status ssh*



ManageNode



ControlNode1



ControlNode2

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

4.1 *sudo ufw allow ssh*

```
jozette@ManageNode:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

ManageNode

```
jozette@controlNode1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

ControlNode1

```
jozette@ControlNode2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

ControlNode2

4.2 *sudo ufw enable*

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

ManageNode

```
jozette@controlNode1:~$ sudo ufw enable
sFirewall is active and enabled on system startup
```

ControlNode1

```
jozette@ControlNode2:~$ sudo ufw enable
sFirewall is active and enabled on system startup
```

ControlNode2

4.3 *sudo ufw status*

```
jozette@ManageNode:~$ sudo ufw status
Status: active

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

ManageNode

```
jozette@controlNode1:~$ sudo ufw status
Status: active

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

ControlNode1

```
jozette@ControlNode2:~$ sudo ufw status
Status: active

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

ControlNode2

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: 192.168.56.1

```
jozette@ManageNode:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.1  netmask 0.0.0.0  broadcast 255.255.255.255
```

1.2 Server 2 IP address: 192.168.56.2

```
jozette@controlNode1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.2  netmask 0.0.0.0  broadcast 255.255.255.255
```

1.3 Server 3 IP address: 192.168.56.3

```
jozette@ControlNode2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.3  netmask 0.0.0.0  broadcast 255.255.255.255
```

2. Make sure that they can ping each other.

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

```
jozette@ManageNode:~$ 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=1.25 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.447 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.59 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.494 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.487 ms
```

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

```
jozette@ManageNode:~$ 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=1.09 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.594 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=2.15 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.565 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.746 ms
```

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

```
jozette@controlNode1:~$ 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=1.65 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=1.87 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.877 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=1.53 ms
```

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

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

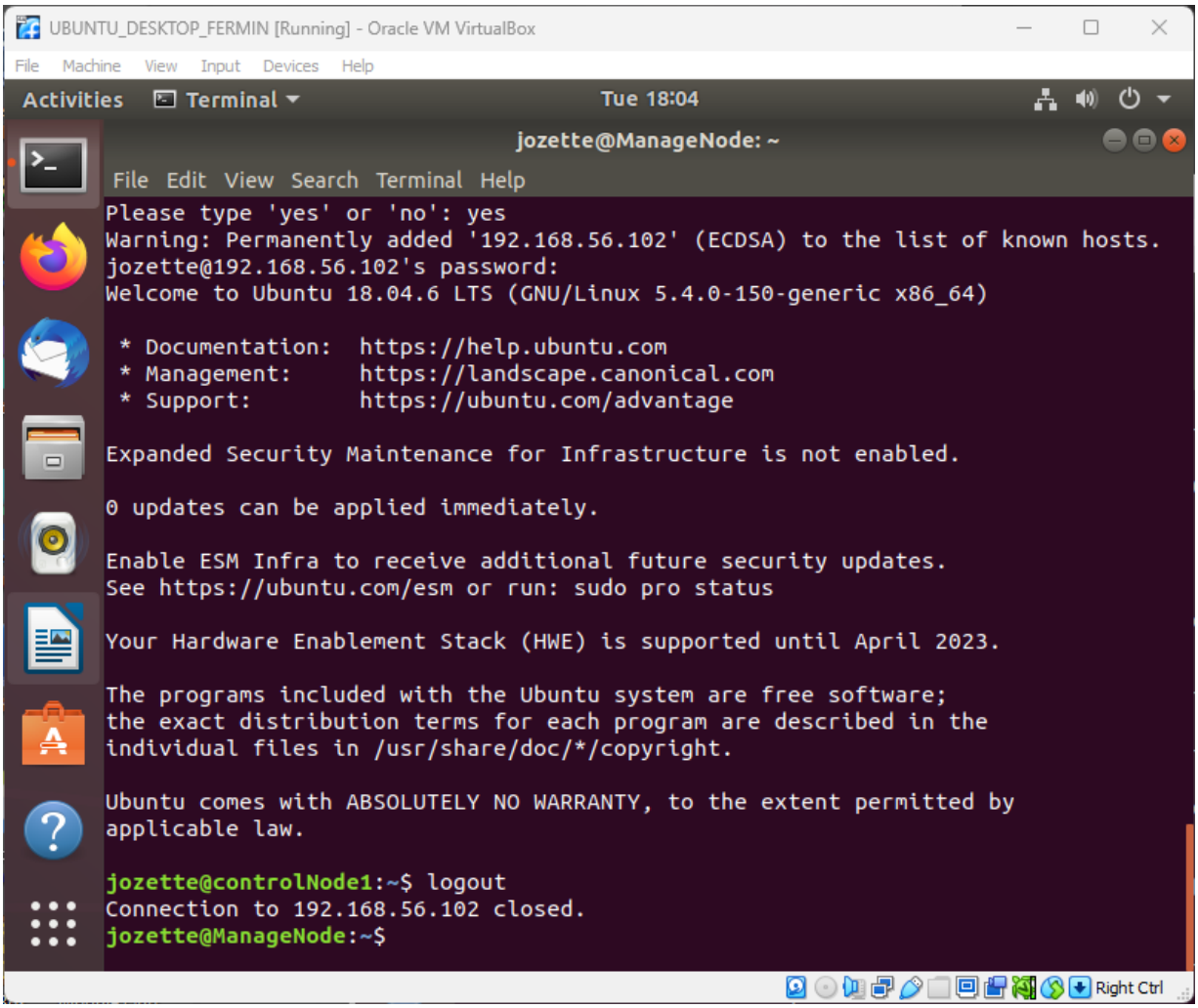
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted

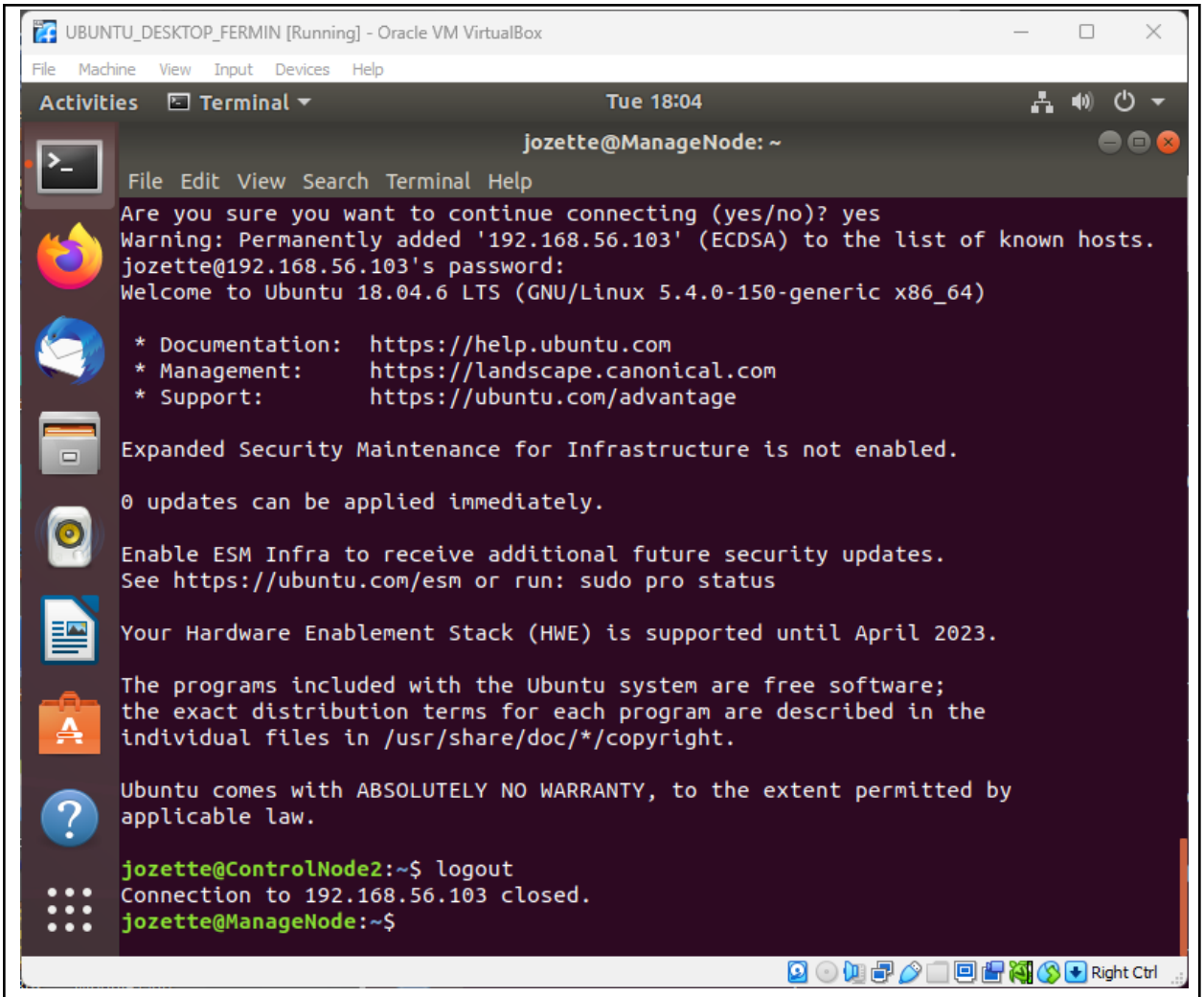
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

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

3. Do the same for Server 2.



controlNode1



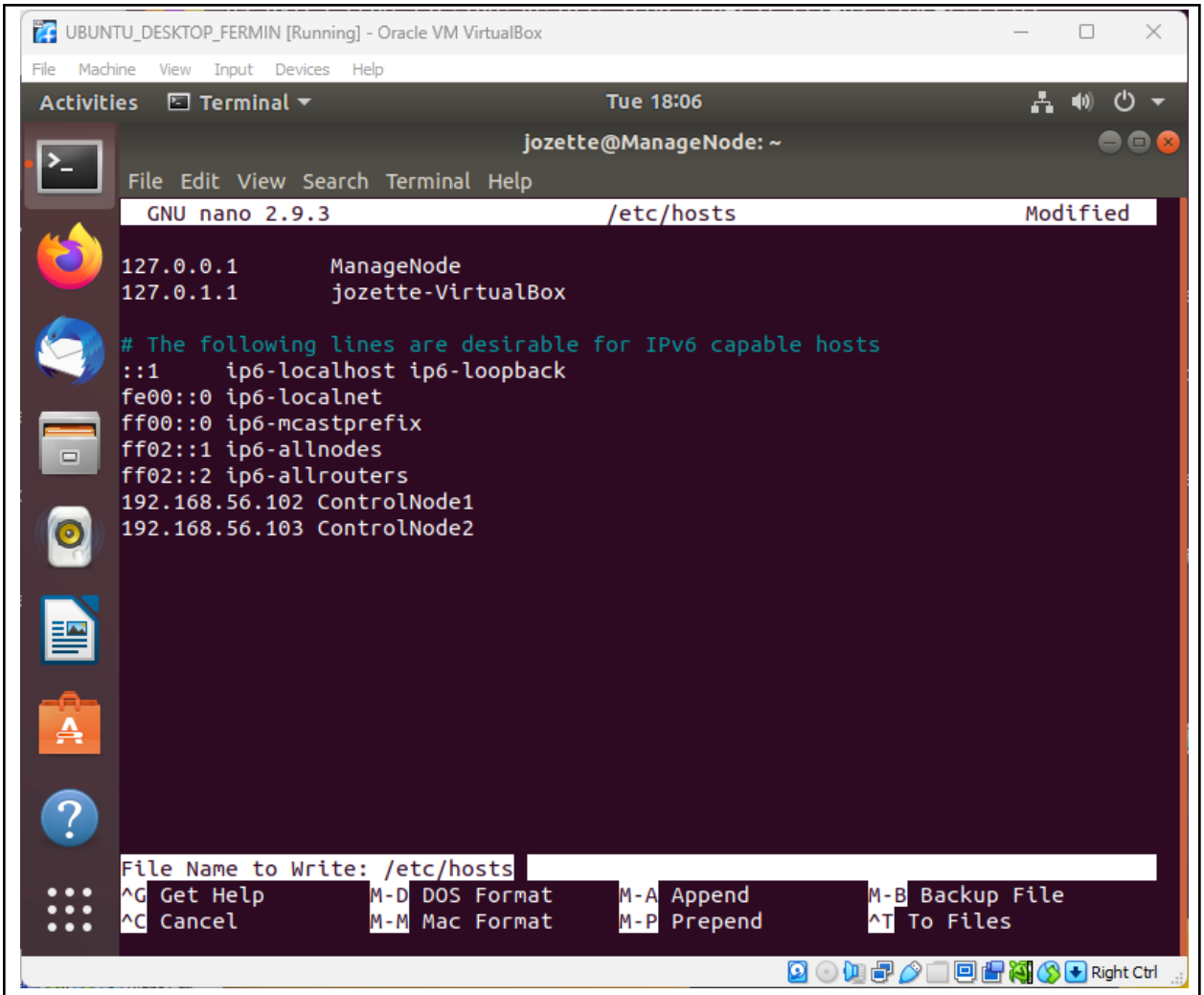
controlNode2

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

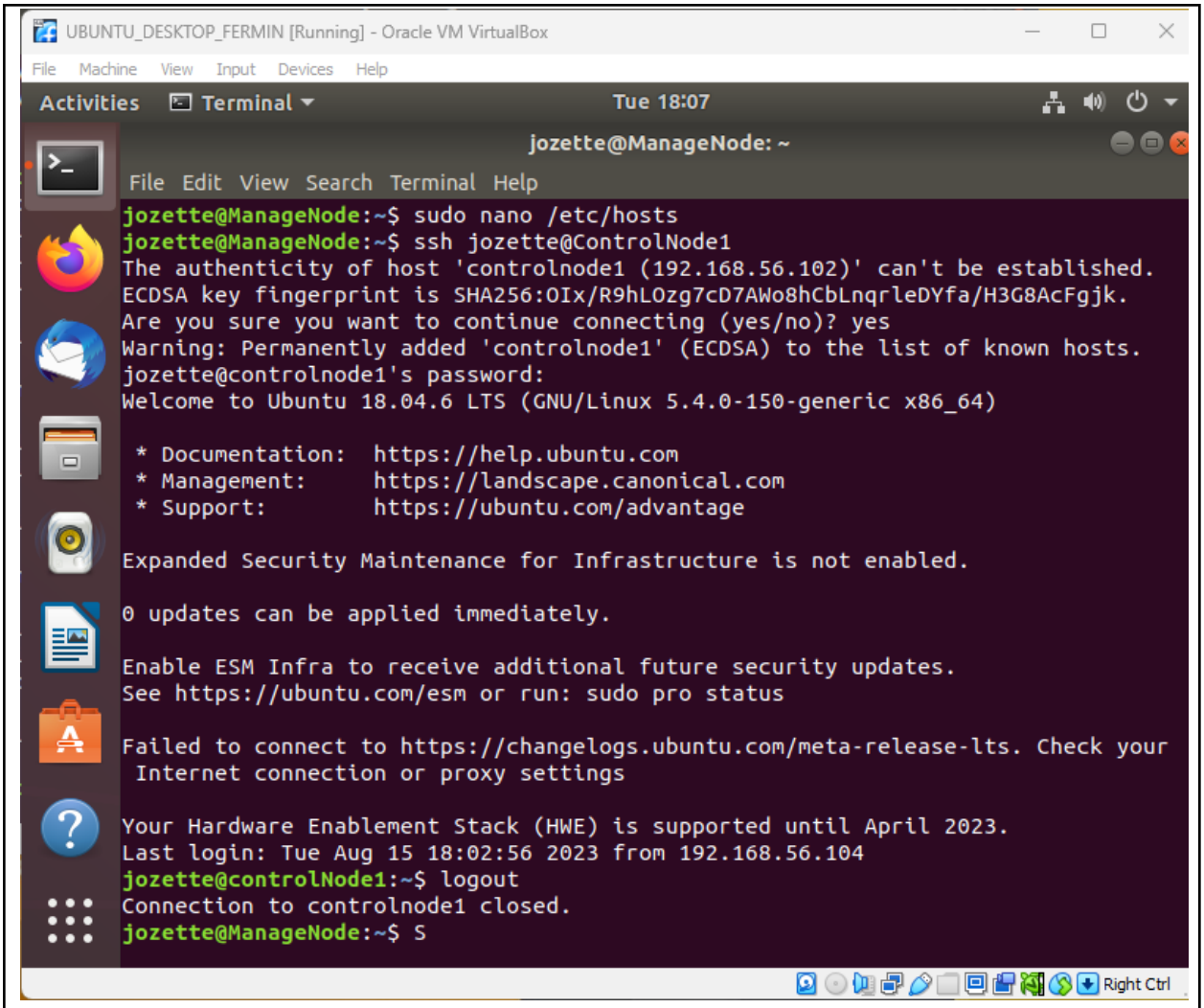
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

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

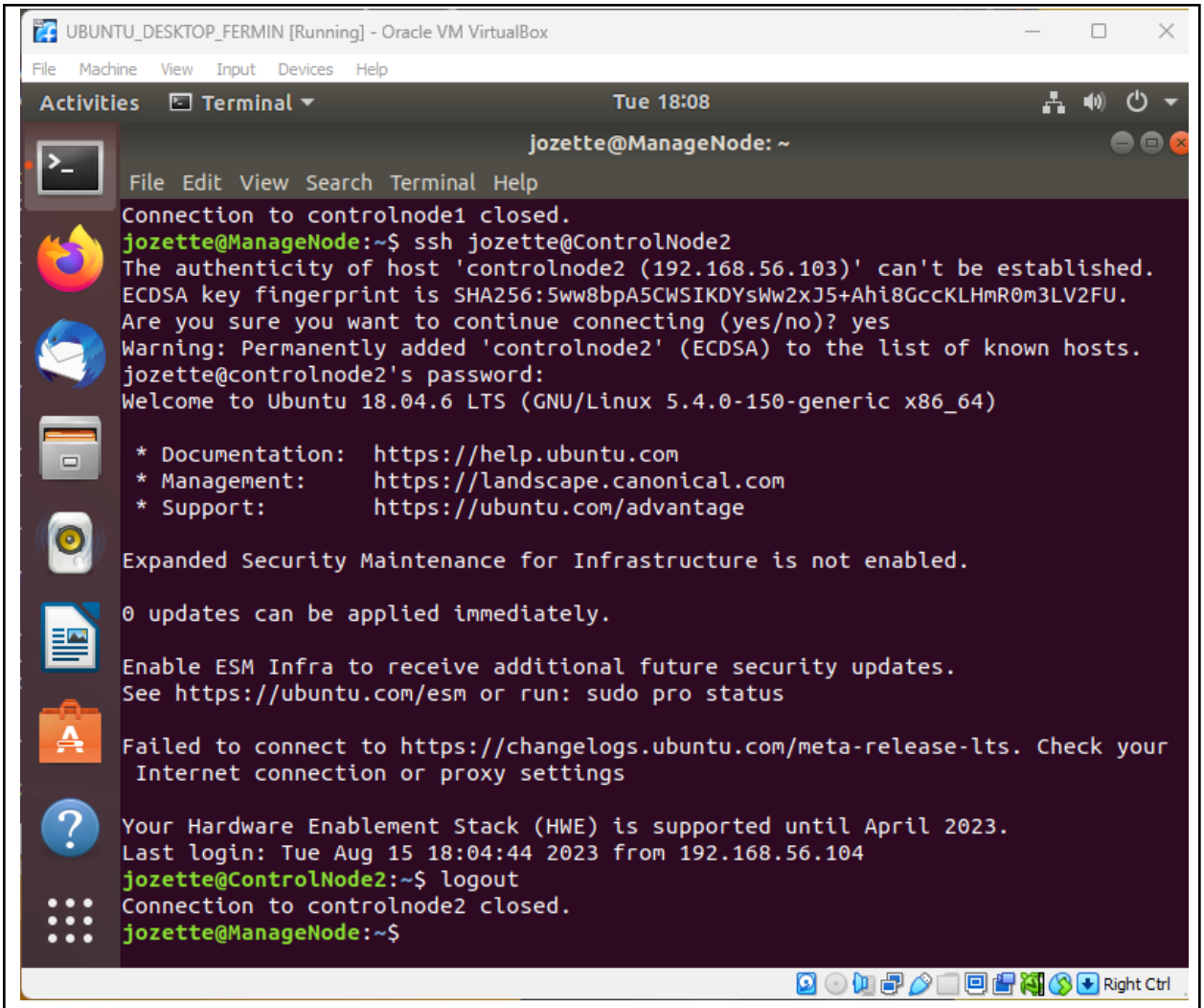
4.3 Save the file and exit.



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.



ControlNode1



ControlNode2

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   we are able to use the hostname instead of IP address in SSH commands because You can just use the hostname command, to make it easier. Now that you have your hostname, add . local to it. This works because it's on your local network.
2. How secured is SSH?

   SSH traffic is secured because it's entirely encrypted. Users' actions are private whether they are sharing a file, browsing the web, or executing a command. While a standard user ID and password can be used to access SSH, public key pairs are more frequently used to authenticate hosts to one another.

Conclusion:

In this hands-on activity I learned how to create and configure networks using a local machine with 2 different servers in ubuntu on virtual box. I also learned why hostnames are able to be used instead of ip addresses in ssh commands. I also learned how secure SSH is. Overall, this hands-on activity is very helpful in learning how to configure a network.