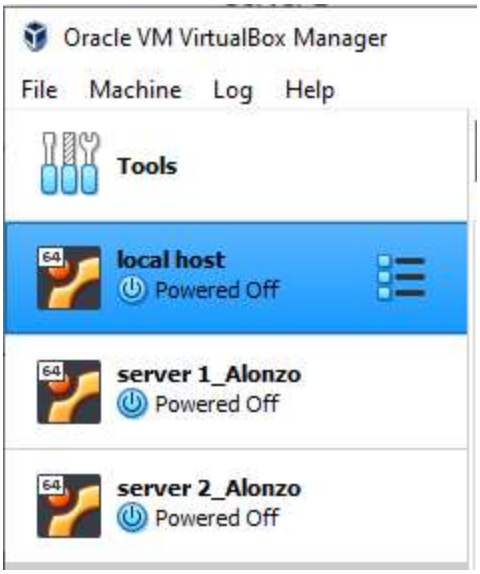
Name: Jozshua Amiel A. Alonzo	Date Performed: August 18, 2022
Course/Section: CPE31S23-A22	Date Submitted: August 18, 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022 - 2023
Activity 1: Configure Network using Virtual Machines	

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*

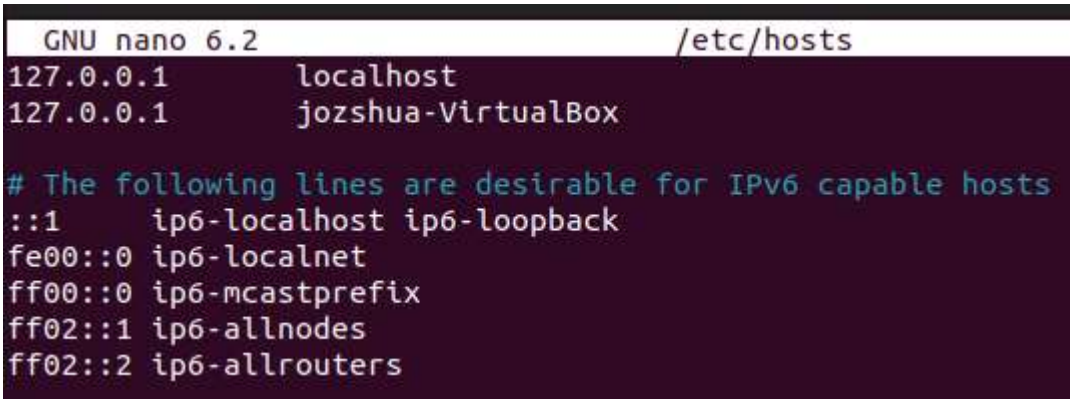
1.1 Use server1 for Server 1

1.2 Use server2 for Server 2

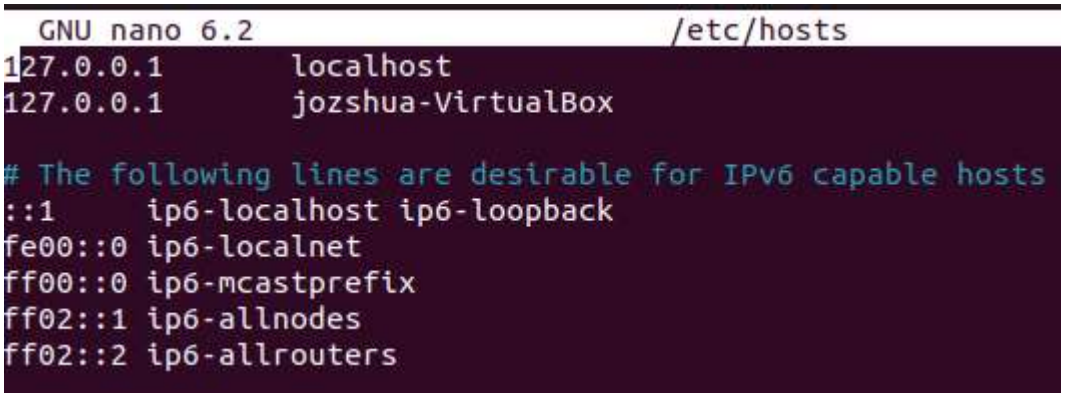
1.3 Use workstation for the Local Machine

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

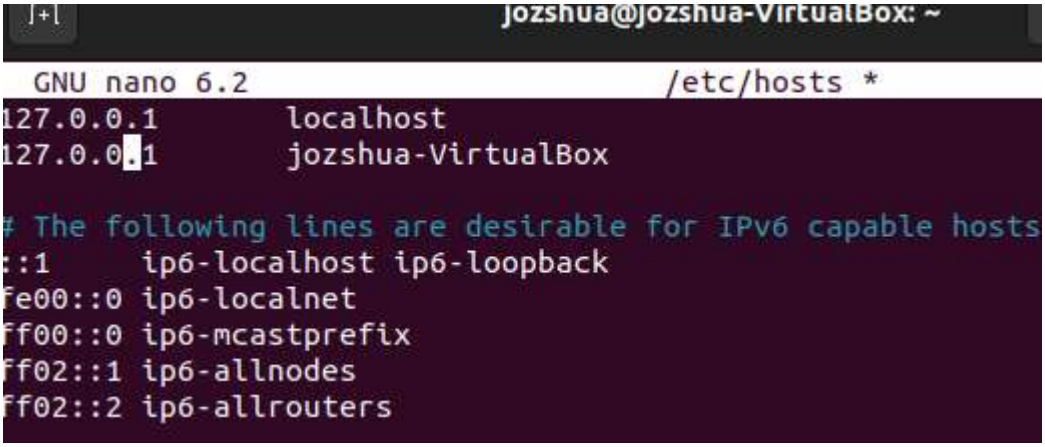
2.1 Type 127.0.0.1 server 1 for Server 1



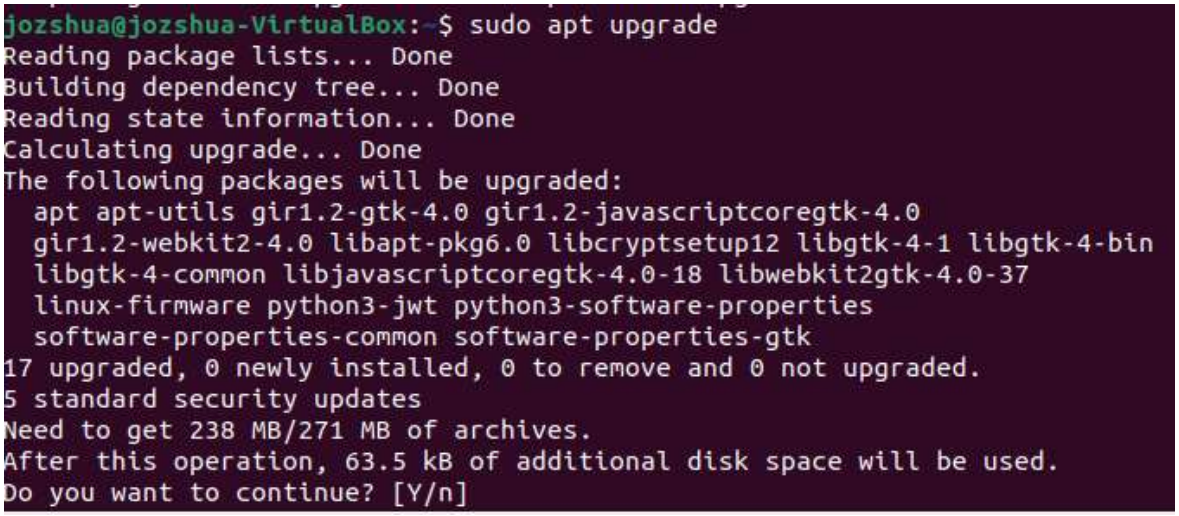
2.2 Type 127.0.0.1 server 2 for Server 2



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.



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

```
jozshua@jozshua-VirtualBox:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
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:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
15 not fully installed or removed.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
E: You don't have enough free space in /var/cache/apt/archives/.
```

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

3.1 *sudo service ssh start*

3.2 *sudo systemctl status ssh*

```
jozshua@server1:~$ sudo service ssh start
jozshua@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Thu 2022-08-18 11:35:24 PST; 2min 57s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 2512 (sshd)
      Tasks: 1 (limit: 1080)
     Memory: 1.7M
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─2512 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 18 11:35:24 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 18 11:35:24 server1 sshd[2512]: Server listening on 0.0.0.0 port 22.
Aug 18 11:35:24 server1 sshd[2512]: Server listening on :: port 22.
Aug 18 11:35:24 server1 systemd[1]: Started OpenBSD Secure Shell server.
```

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

4.1 *sudo ufw allow ssh*

4.2 *sudo ufw enable*

sudo ufw status

```
jozshua@jozshua-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
jozshua@jozshua-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
jozshua@jozshua-VirtualBox:~$ 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

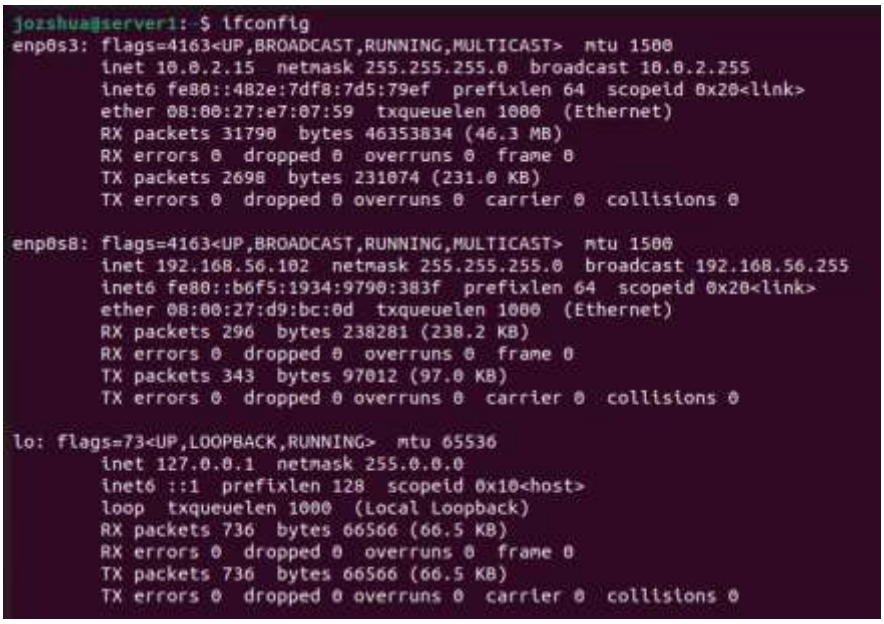
1.2 Server 2 IP address: 192.168.56.103

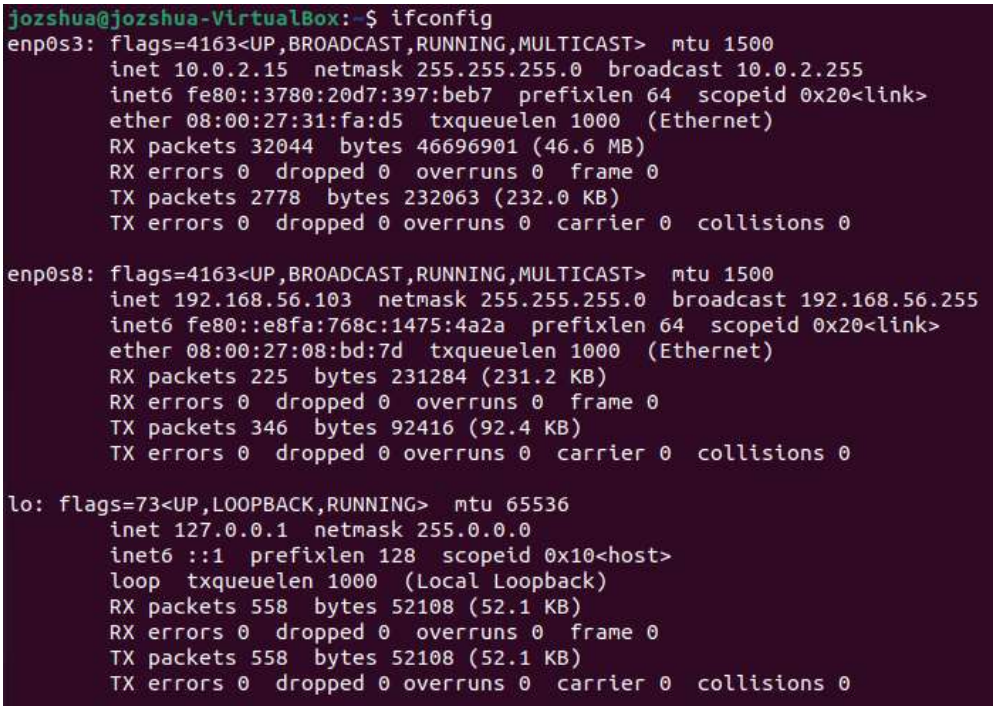
1.3 Server 3 IP address: 192.168.56.___

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 SuccessfulConnectivity test for Server 1 to Server 2: ☐ Successful ☐ Not Successful


jozshua@server1:~$ 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::482e:7df8:7d5:79ef prefixlen 64 scopeid 0x20<link>
ether 08:00:27:e7:07:59 txqueuelen 1000 (Ethernet)
RX packets 31790 bytes 46353834 (46.3 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2698 bytes 231074 (231.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::b6f5:1934:9790:383f prefixlen 64 scopeid 0x20<link>
ether 08:00:27:d9:bc:0d txqueuelen 1000 (Ethernet)
RX packets 296 bytes 238281 (238.2 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 343 bytes 97012 (97.0 KB)
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 736 bytes 66566 (66.5 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 736 bytes 66566 (66.5 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0


jozshua@jozshua-VirtualBox:~$ 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::3780:20d7:397:beb7 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:31:fa:d5 txqueuelen 1000 (Ethernet)
RX packets 32044 bytes 46696901 (46.6 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2778 bytes 232063 (232.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::e8fa:768c:1475:4a2a prefixlen 64 scopeid 0x20<link>
ether 08:00:27:08:bd:7d txqueuelen 1000 (Ethernet)
RX packets 225 bytes 231284 (231.2 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 346 bytes 92416 (92.4 KB)
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 558 bytes 52108 (52.1 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 558 bytes 52108 (52.1 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```
jozshua@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=0.541 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.409 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.17 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.668 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=1.18 ms
^C
--- 192.168.56.103 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5050ms
rtt min/avg/max/mdev = 0.409/0.835/1.179/0.308 ms
```

```
jozshua@jozshua-VirtualBox:~$ 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=0.713 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.594 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=1.16 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.632 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.866 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=1.09 ms
^C
--- 192.168.56.102 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7033ms
rtt min/avg/max/mdev = 0.594/0.901/1.161/0.215 ms
```

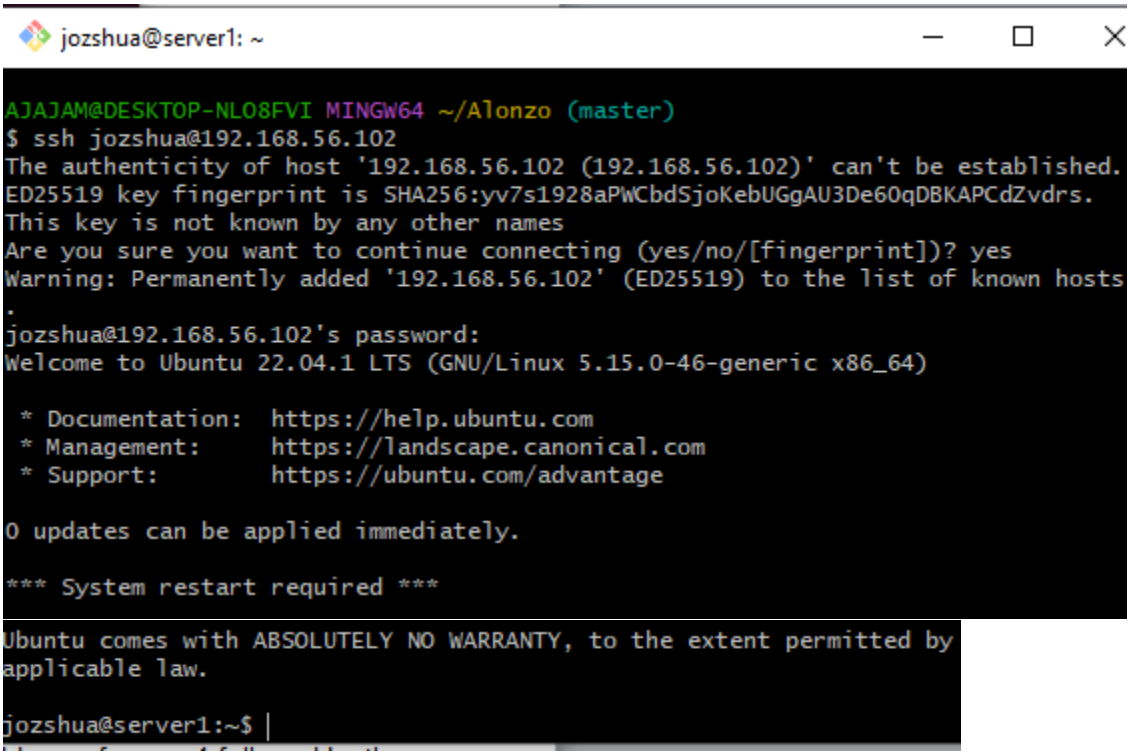
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

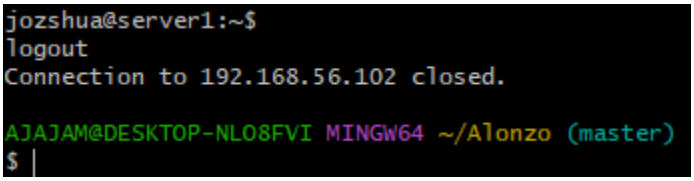
1.1 ssh username@ip_address_server1 for example, *ssh* jvtaylar@192.168.56.120

1.2 Enter the password for server 1 when prompted

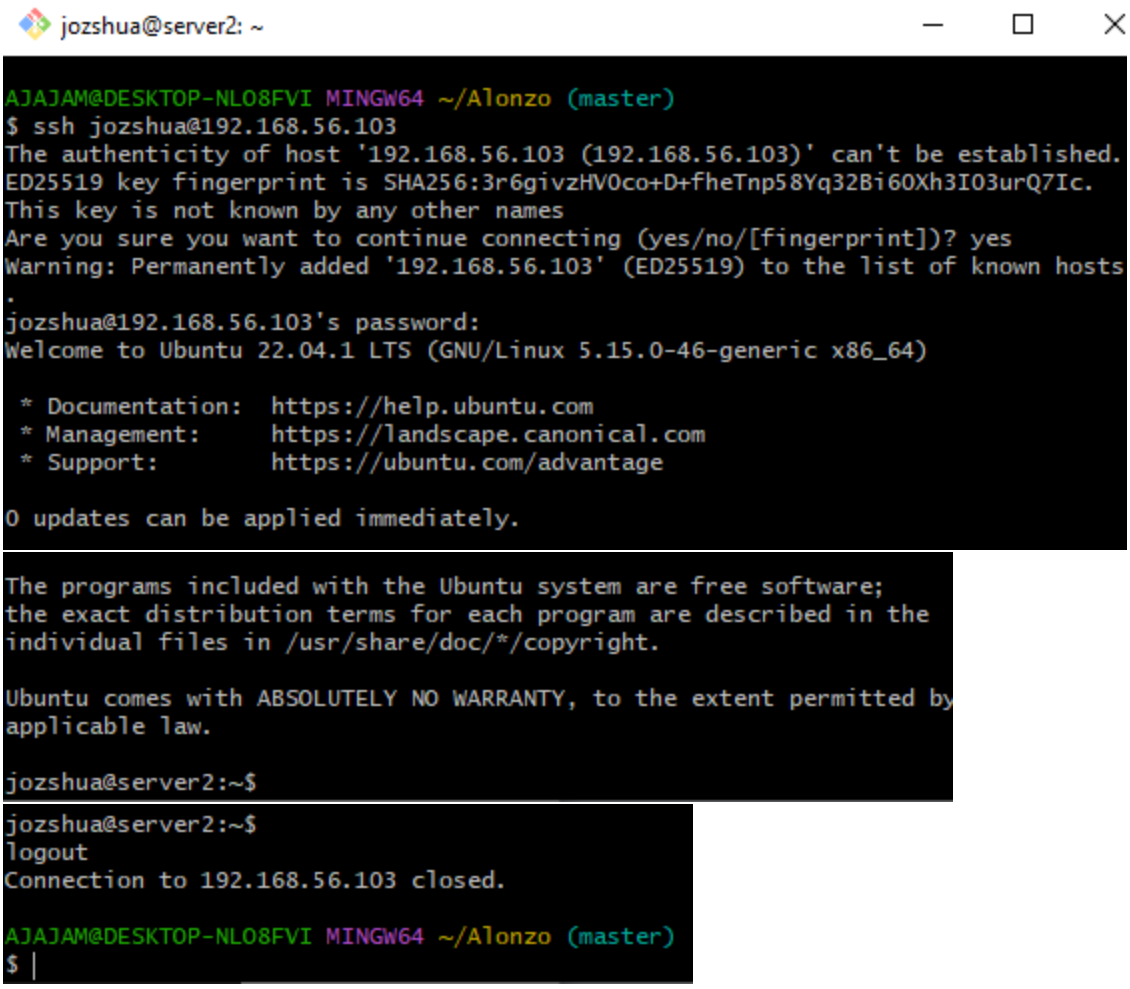
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*



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



3. Do the same for Server 2.

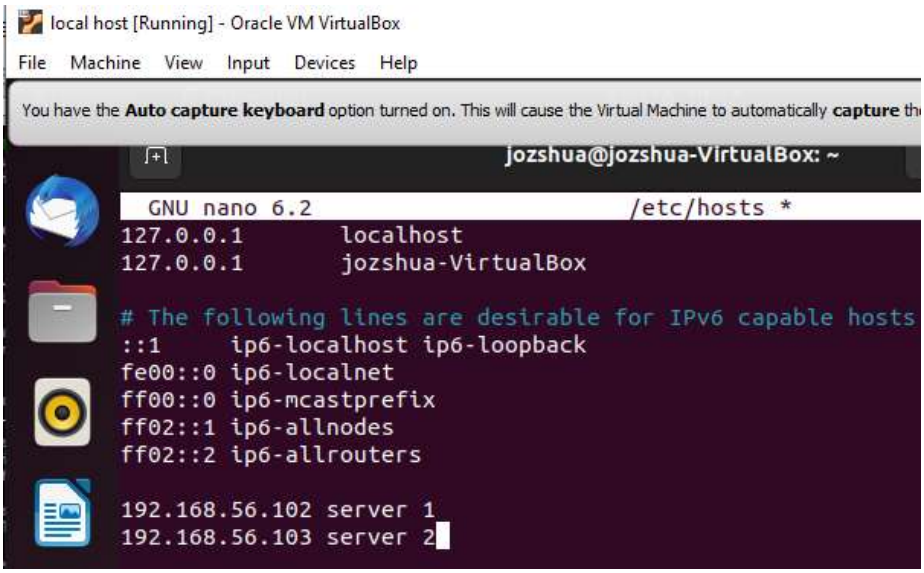


4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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.



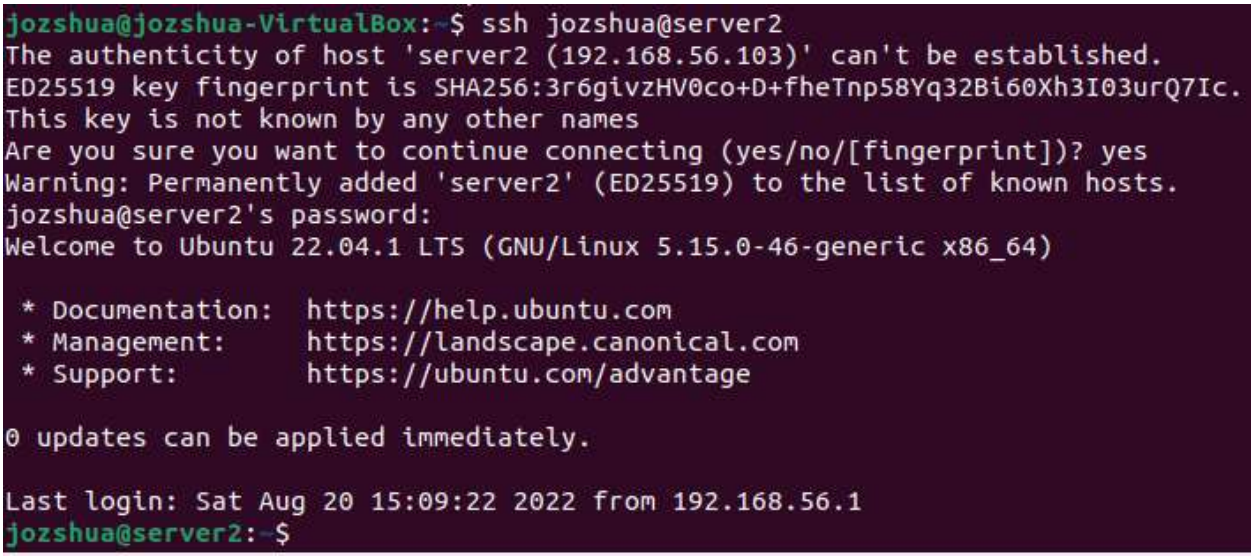
On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

```
jozshua@jozshua-VirtualBox:~$ ssh jozshua@server1
The authenticity of host 'server1 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:yv7s1928aPWCbdSjoKebUGgAU3De60qDBKAPCdZvdrs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server1' (ED25519) to the list of known hosts.
jozshua@server1's password:
jozshua@server1's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

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

0 updates can be applied immediately.

*** System restart required ***
Last login: Sat Aug 20 14:32:17 2022 from 192.168.56.1
jozshua@server1:~$
```



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 hostname in SSH command because of sudo nano /etc/hostname command that we did in the local host. By typing it on the inside the ip address and together with the hostname in 1 line and that's it. Then we call from the command ssh hostname@server#.

2. How secured is SSH?

   The SSH is secured because it can secure the connection between a client and in the user's command, authentication, and the attacks and access from the unauthorized user.