Name: Pacios, Angela Monique A.	**Date Performed:** 08-14-23
Course/Section: CPE232 - CPE31S4	**Date Submitted:** 08-15-23
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Semester ‘23 - ‘24

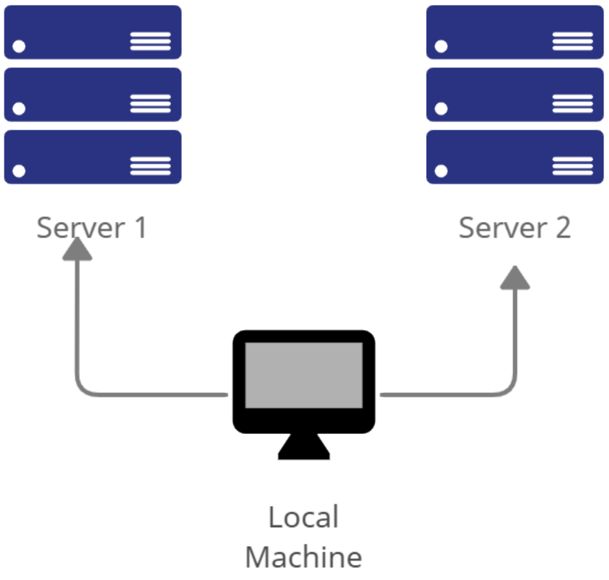
Activity 1: Configure Network using Virtual Machines

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:**

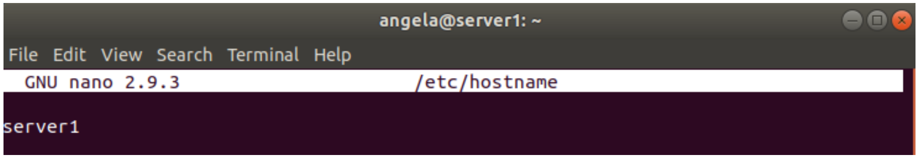
Network Topology:

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*).



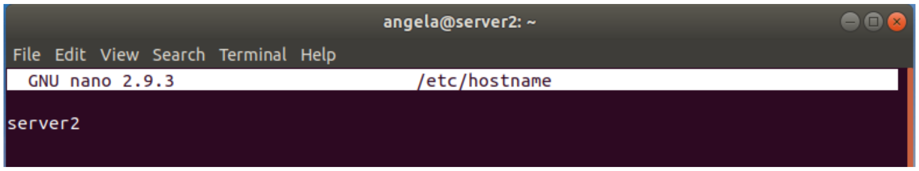
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1



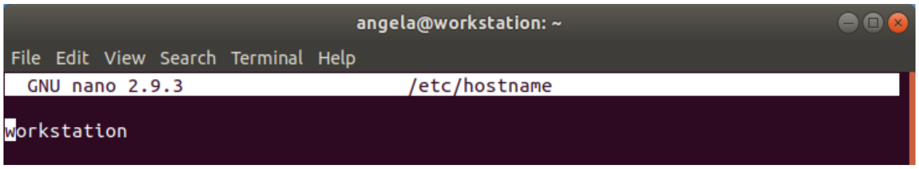
```
  GNU nano 2.9.3                   /etc/hostname

server1
```

1.2 Use server2 for Server 2



```
  GNU nano 2.9.3                   /etc/hostname

server2
```

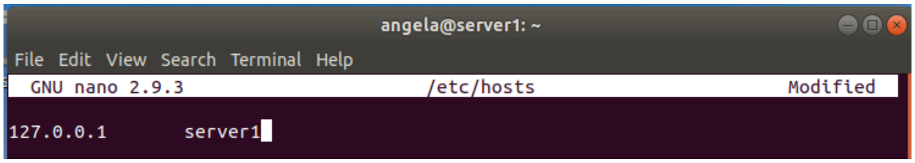
1.3 Use workstation for the Local Machine



```
  GNU nano 2.9.3                  /etc/hostname

workstation
```

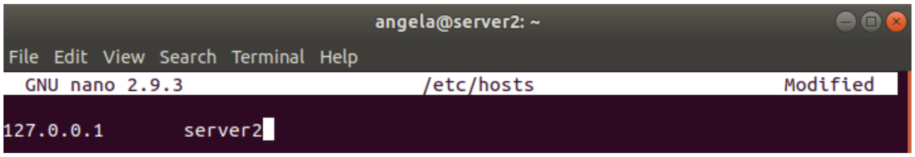
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

2.1 Type 127.0.0.1 server 1 for Server 1



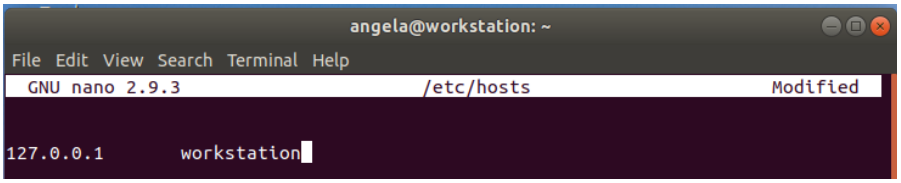
```
  GNU nano 2.9.3                   /etc/hosts                      Modified

127.0.0.1       server1
```

2.2 Type 127.0.0.1 server 2 for Server 2



```
  GNU nano 2.9.3                   /etc/hosts                      Modified

127.0.0.1       server2
```

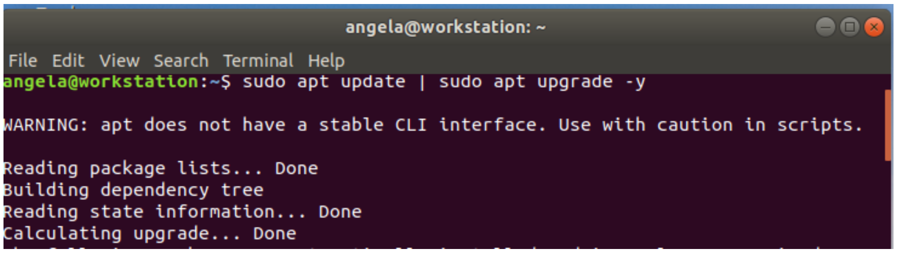
2.3 Type 127.0.0.1 workstation for the Local Machine



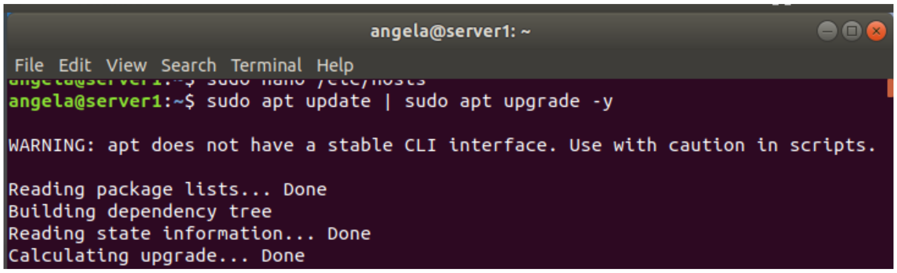
```
  GNU nano 2.9.3                   /etc/hosts                    Modified


127.0.0.1       workstation
```

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

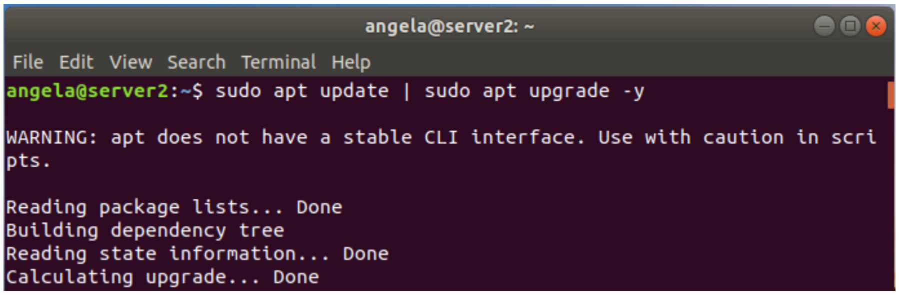
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.



```
angela@workstation:~$ sudo apt update | sudo apt upgrade -y

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
```

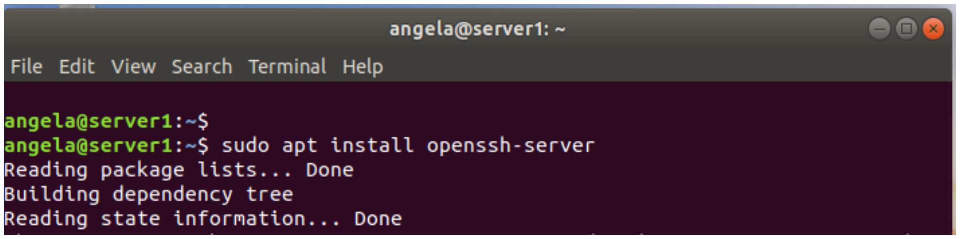


```
angela@server1:~$ sudo nano /etc/hosts
angela@server1:~$ sudo apt update | sudo apt upgrade -y

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
```

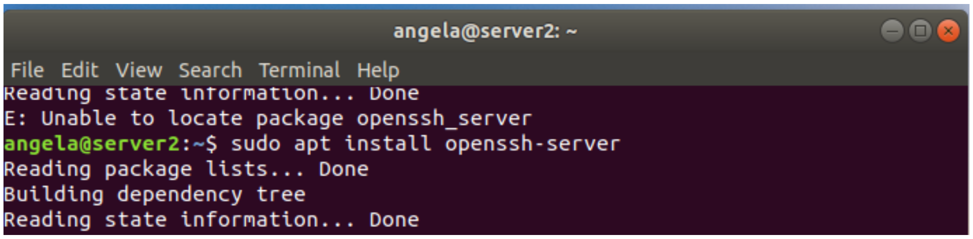


```
angela@server2:~$ sudo apt update | sudo apt upgrade -y

WARNING: apt does not have a stable CLI interface. Use with caution in scri
pts.

Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
```

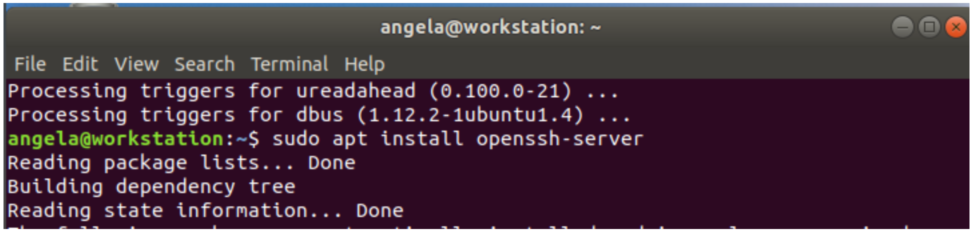
2. Install the SSH server using the command *sudo apt install openssh-server*.



```

angela@server1:~$
angela@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```



```
Reading state information... Done
E: Unable to locate package openssh_server
angela@server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

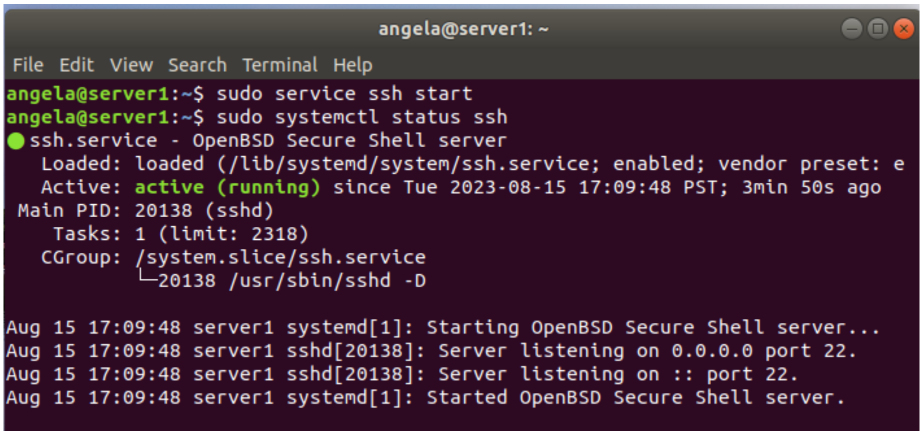


```
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for dbus (1.12.2-1ubuntu1.4) ...
angela@workstation:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

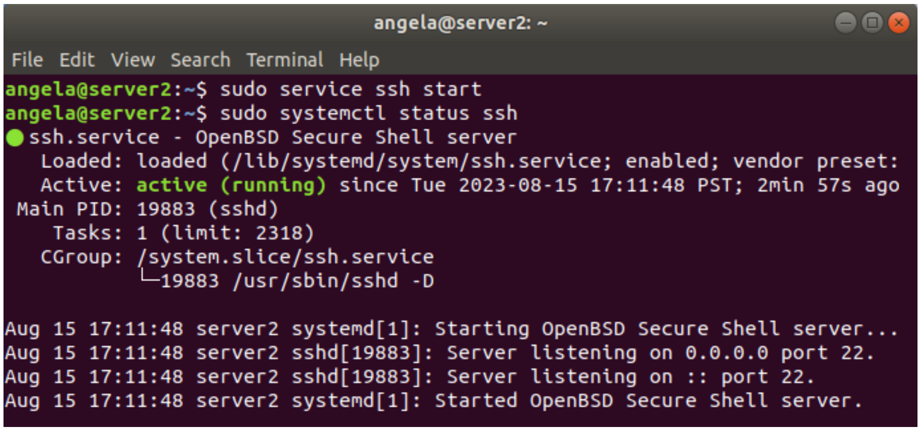
3. Verify if the SSH service has started by issuing the following commands:

   3.1 *sudo service ssh start*

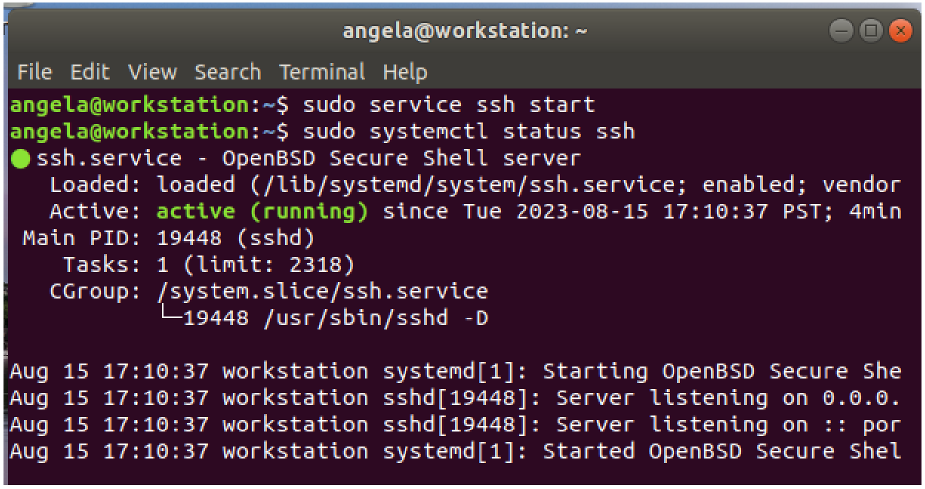
   3.2 *sudo systemctl status ssh*



```
angela@server1:~$ sudo service ssh start
angela@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e
   Active: active (running) since Tue 2023-08-15 17:09:48 PST; 3min 50s ago
 Main PID: 20138 (sshd)
    Tasks: 1 (limit: 2318)
   CGroup: /system.slice/ssh.service
           └─20138 /usr/sbin/sshd -D

Aug 15 17:09:48 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 15 17:09:48 server1 sshd[20138]: Server listening on 0.0.0.0 port 22.
Aug 15 17:09:48 server1 sshd[20138]: Server listening on :: port 22.
Aug 15 17:09:48 server1 systemd[1]: Started OpenBSD Secure Shell server.
```



```
angela@server2:~$ sudo service ssh start
angela@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   Active: active (running) since Tue 2023-08-15 17:11:48 PST; 2min 57s ago
 Main PID: 19883 (sshd)
    Tasks: 1 (limit: 2318)
   CGroup: /system.slice/ssh.service
           └─19883 /usr/sbin/sshd -D

Aug 15 17:11:48 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 15 17:11:48 server2 sshd[19883]: Server listening on 0.0.0.0 port 22.
Aug 15 17:11:48 server2 sshd[19883]: Server listening on :: port 22.
Aug 15 17:11:48 server2 systemd[1]: Started OpenBSD Secure Shell server.
```



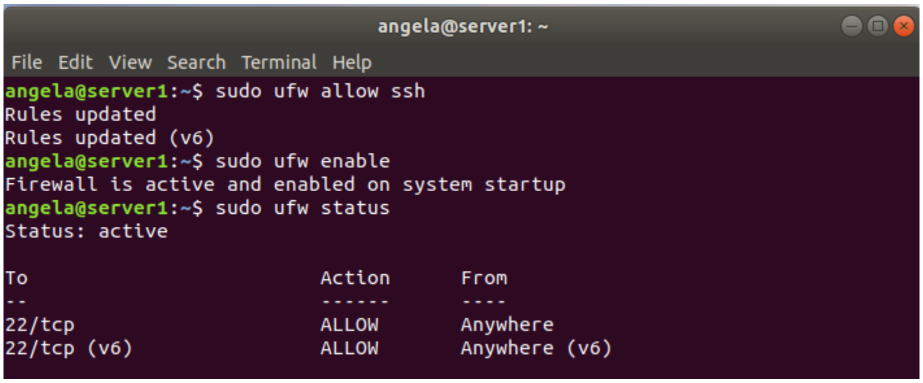
```
angela@workstation:~$ sudo service ssh start
angela@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor
   Active: active (running) since Tue 2023-08-15 17:10:37 PST; 4min
 Main PID: 19448 (sshd)
    Tasks: 1 (limit: 2318)
   CGroup: /system.slice/ssh.service
           └─19448 /usr/sbin/sshd -D

Aug 15 17:10:37 workstation systemd[1]: Starting OpenBSD Secure She
Aug 15 17:10:37 workstation sshd[19448]: Server listening on 0.0.0.
Aug 15 17:10:37 workstation sshd[19448]: Server listening on :: por
Aug 15 17:10:37 workstation systemd[1]: Started OpenBSD Secure Shel
```

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

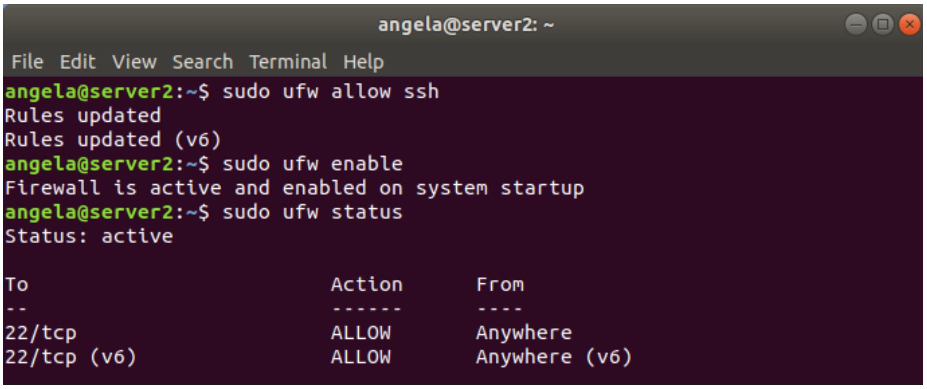
   ***4.1** sudo ufw allow ssh*

   ***4.2** sudo ufw enable*

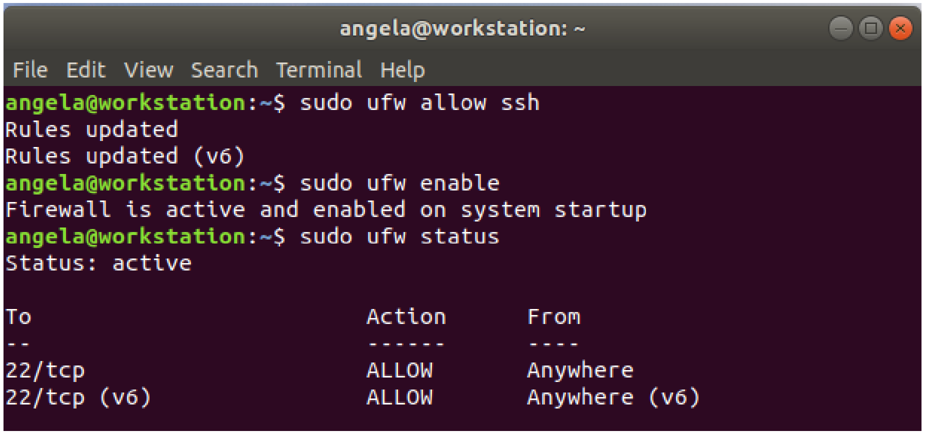
   ***4.3** sudo ufw status*



```
angela@server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
angela@server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
angela@server1:~$ sudo ufw status
Status: active

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



```
angela@server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
angela@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
angela@server2:~$ sudo ufw status
Status: active

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

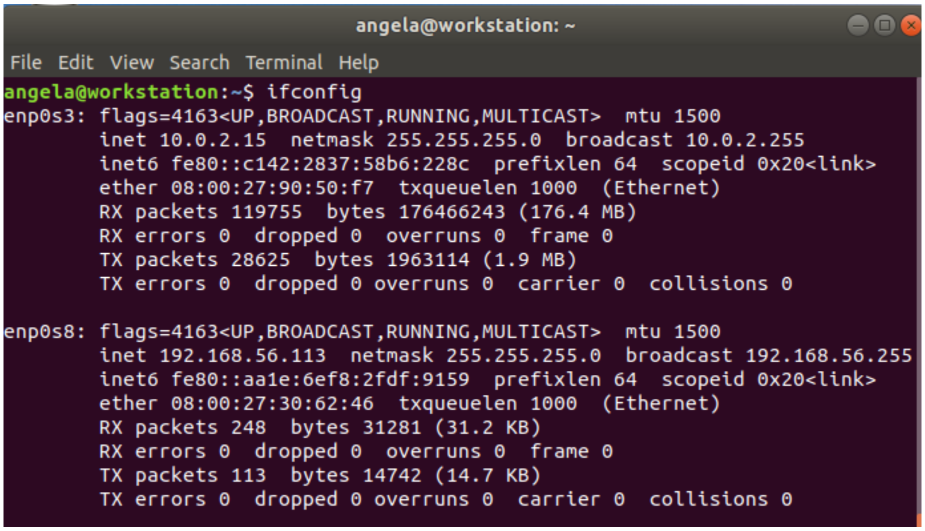


```
angela@workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
angela@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
angela@workstation:~$ sudo ufw status
Status: active

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:

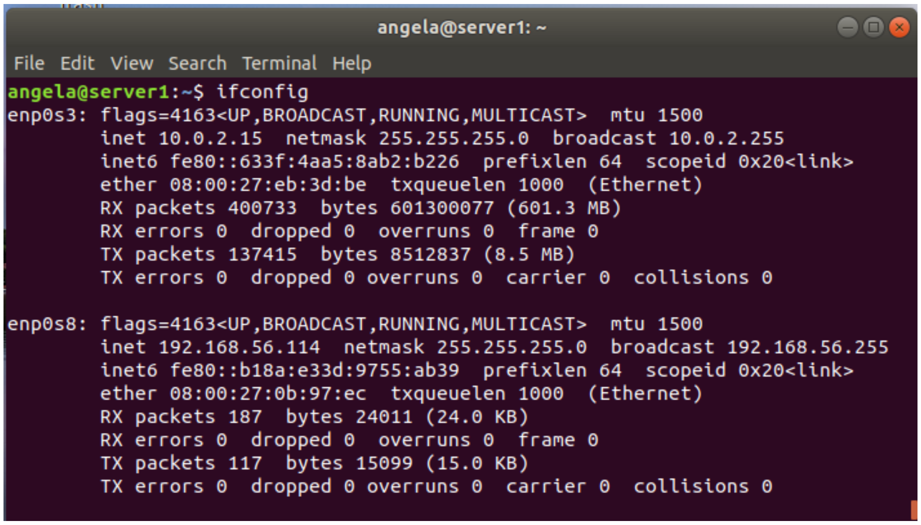
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.

1.1 workstation IP address: 192.168.56.**113**



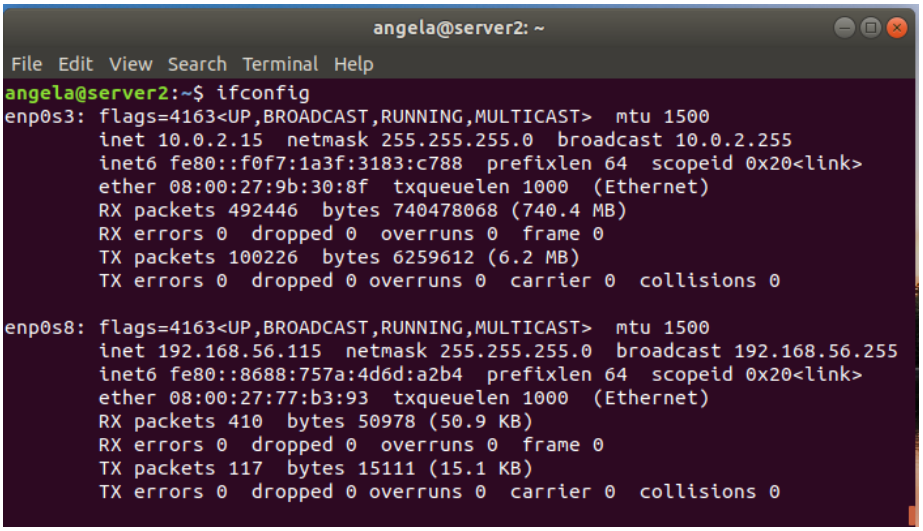
```
angela@workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::c142:2837:58b6:228c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:90:50:f7  txqueuelen 1000  (Ethernet)
        RX packets 119755  bytes 176466243 (176.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28625  bytes 1963114 (1.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.113  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::aa1e:6ef8:2fdf:9159  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:30:62:46  txqueuelen 1000  (Ethernet)
        RX packets 248  bytes 31281 (31.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 113  bytes 14742 (14.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

1.2 Server 1 IP address: 192.168.56.**114**



```
angela@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::633f:4aa5:8ab2:b226  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:eb:3d:be  txqueuelen 1000  (Ethernet)
        RX packets 400733  bytes 601300077 (601.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 137415  bytes 8512837 (8.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.114  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::b18a:e33d:9755:ab39  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0b:97:ec  txqueuelen 1000  (Ethernet)
        RX packets 187  bytes 24011 (24.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 15099 (15.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

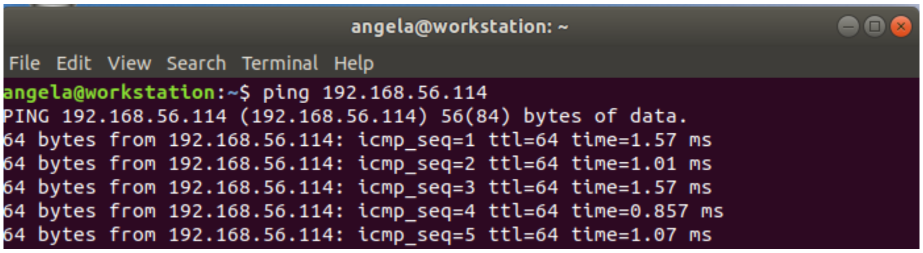
1.3 Server 2 IP address: 192.168.56.**115**



```
angela@server2:~$ 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::f0f7:1a3f:3183:c788  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9b:30:8f  txqueuelen 1000  (Ethernet)
        RX packets 492446  bytes 740478068 (740.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 100226  bytes 6259612 (6.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.115  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::8688:757a:4d6d:a2b4  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:77:b3:93  txqueuelen 1000  (Ethernet)
        RX packets 410  bytes 50978 (50.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 15111 (15.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

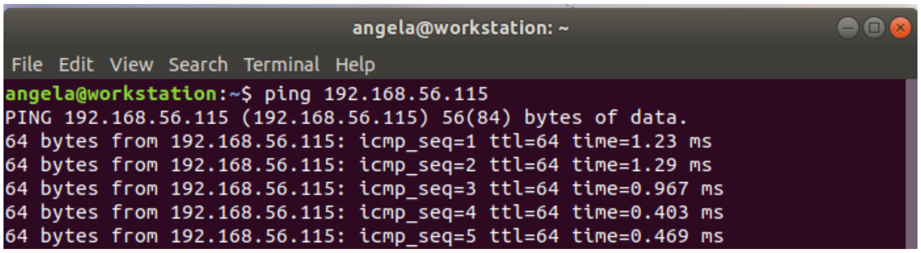
2. Make sure that they can ping each other.

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



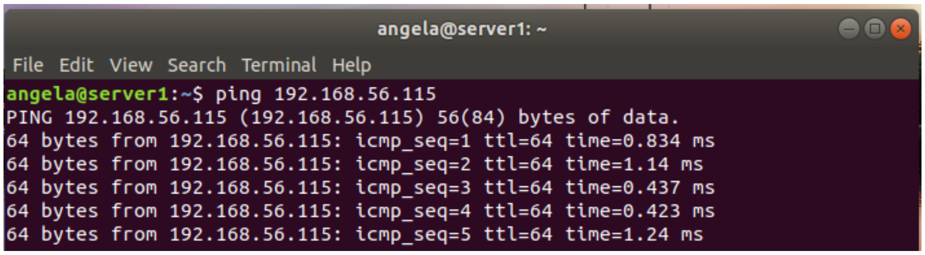
```
angela@workstation:~$ ping 192.168.56.114
PING 192.168.56.114 (192.168.56.114) 56(84) bytes of data.
64 bytes from 192.168.56.114: icmp_seq=1 ttl=64 time=1.57 ms
64 bytes from 192.168.56.114: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 192.168.56.114: icmp_seq=3 ttl=64 time=1.57 ms
64 bytes from 192.168.56.114: icmp_seq=4 ttl=64 time=0.857 ms
64 bytes from 192.168.56.114: icmp_seq=5 ttl=64 time=1.07 ms
```

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



```
angela@workstation:~$ ping 192.168.56.115
PING 192.168.56.115 (192.168.56.115) 56(84) bytes of data.
64 bytes from 192.168.56.115: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from 192.168.56.115: icmp_seq=2 ttl=64 time=1.29 ms
64 bytes from 192.168.56.115: icmp_seq=3 ttl=64 time=0.967 ms
64 bytes from 192.168.56.115: icmp_seq=4 ttl=64 time=0.403 ms
64 bytes from 192.168.56.115: icmp_seq=5 ttl=64 time=0.469 ms
```

2.3 Connectivity test for Server 1 to Server 2: ☑ Successful □ Not Successful



```
angela@server1:~$ ping 192.168.56.115
PING 192.168.56.115 (192.168.56.115) 56(84) bytes of data.
64 bytes from 192.168.56.115: icmp_seq=1 ttl=64 time=0.834 ms
64 bytes from 192.168.56.115: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from 192.168.56.115: icmp_seq=3 ttl=64 time=0.437 ms
64 bytes from 192.168.56.115: icmp_seq=4 ttl=64 time=0.423 ms
64 bytes from 192.168.56.115: icmp_seq=5 ttl=64 time=1.24 ms
```

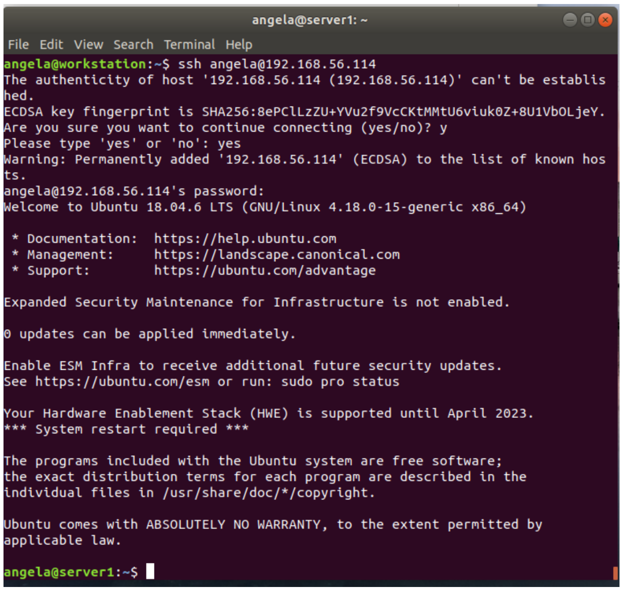
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

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

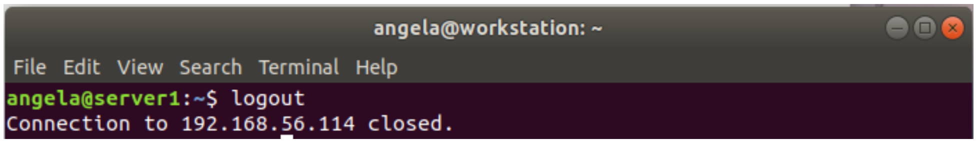
1.2 Enter the password for server 1 when prompted

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



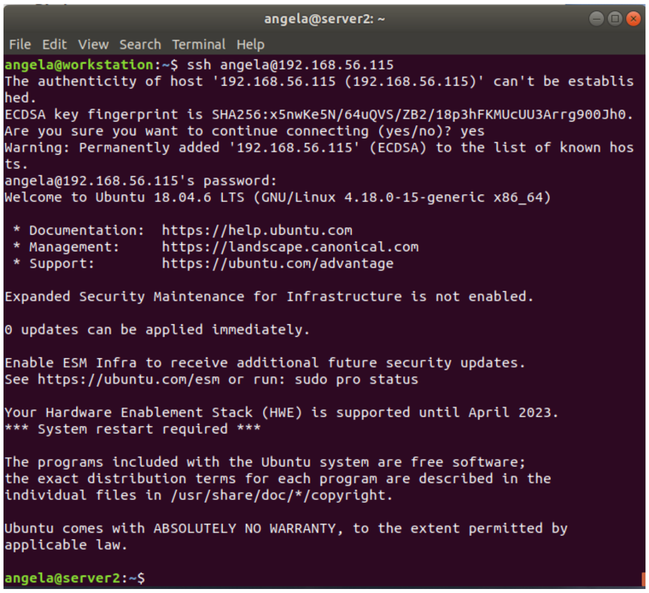
```
angela@workstation:~$ ssh angela@192.168.56.114
The authenticity of host '192.168.56.114 (192.168.56.114)' can't be establis
hed.
ECDSA key fingerprint is SHA256:8ePClLzZU+YVu2f9VcCKtMMtU6viuk0Z+8U1VbOLjeY.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.56.114' (ECDSA) to the list of known hos
ts.
angela@192.168.56.114's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

angela@server1:~$ 
```

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

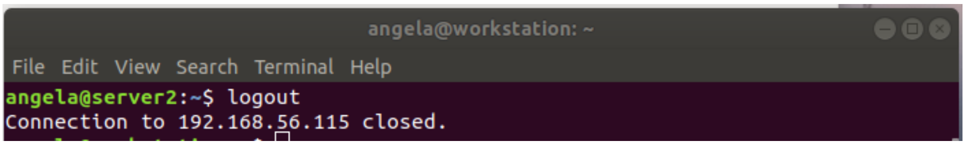


```
angela@server1:~$ logout
Connection to 192.168.56.114 closed.
```

3. Do the same for Server 2.

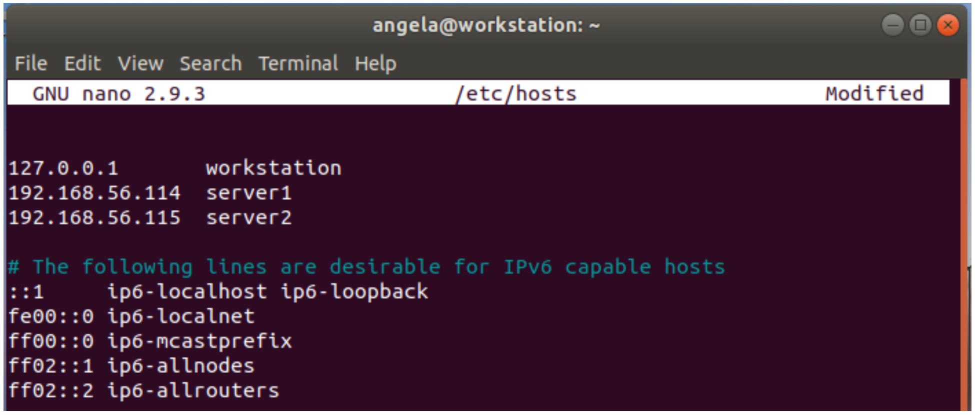


```
angela@workstation:~$ ssh angela@192.168.56.115
The authenticity of host '192.168.56.115 (192.168.56.115)' can't be establis
hed.
ECDSA key fingerprint is SHA256:x5nwKe5N/64uQVS/ZB2/18p3hFKMUcUU3Arrg900Jh0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.115' (ECDSA) to the list of known hos
ts.
angela@192.168.56.115's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

angela@server2:~$
```



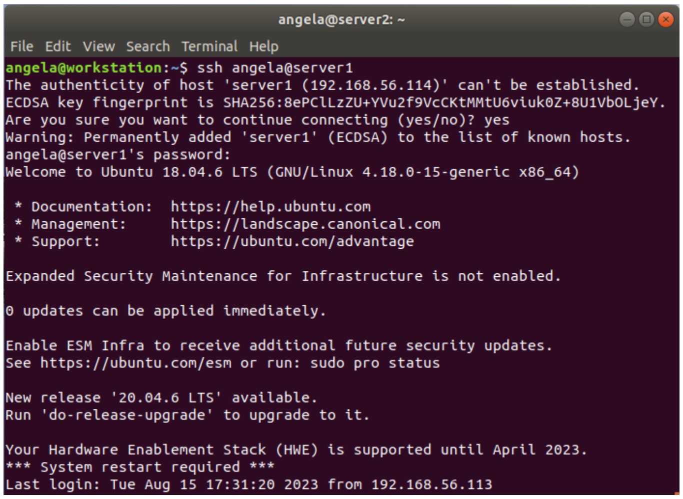
```
angela@server2:~$ logout
Connection to 192.168.56.115 closed.
```

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.

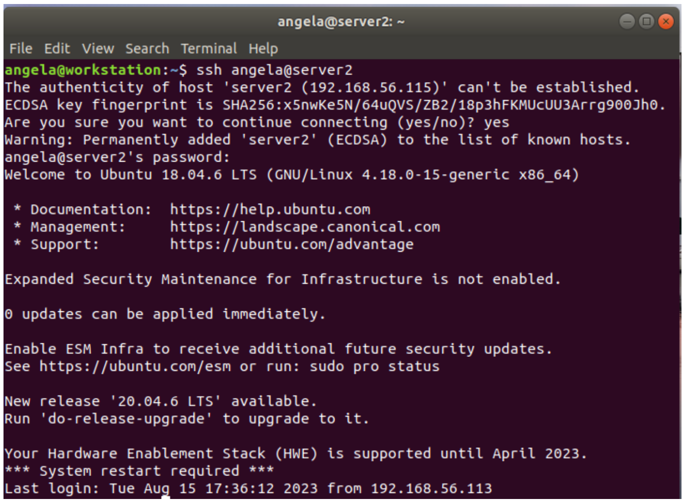


```
  GNU nano 2.9.3                    /etc/hosts                    Modified


127.0.0.1       workstation
192.168.56.114  server1
192.168.56.115  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
```

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.



```
angela@workstation:~$ ssh angela@server1
The authenticity of host 'server1 (192.168.56.114)' can't be established.
ECDSA key fingerprint is SHA256:8ePClLzZU+YVu2f9VcCKtMMtU6viuk0Z+8U1VbOLjeY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1' (ECDSA) to the list of known hosts.
angela@server1's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***
Last login: Tue Aug 15 17:31:20 2023 from 192.168.56.113
```



```
angela@workstation:~$ ssh angela@server2
The authenticity of host 'server2 (192.168.56.115)' can't be established.
ECDSA key fingerprint is SHA256:x5nwKe5N/64uQVS/ZB2/18p3hFKMUcUU3Arrg900Jh0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server2' (ECDSA) to the list of known hosts.
angela@server2's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***
Last login: Tue Aug 15 17:36:12 2023 from 192.168.56.113
```

Reflections:

Answer the following:

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

    We were able to connect and use the SSH commands once we edited the ip address and assigned a hostname for it. This was possible when we edited the hostname using the sudo nano /etc/hostname and sudo nano /etc/hosts for the hostname and the ip address.

2. How secure is SSH?

    SSH is very secure as it encrypts keys and requires authentication for passwords and other sensitive content. Multiple authentication is also needed for logins especially for public or local computers. This allows more security where the contents are not easily seen and modified.

Conclusion:

For this very first activity for this course, it was quite easy as the instructions were very clear and other components through it were already learned for the previous related course. Firstly, we created our own Ubuntu desktop and modified it to be able to clone and connect to different servers. After the cloning process was finished, we edited the hostname for the different Ubuntu desktop. This hostname that we have assigned also serves as our connecting hostnames for the different servers if we want to connect to them. There are more commands to make it possible and connecting the servers to each other but seeing it works at the end is quite joyful. Connecting and the ping proves that the desktops / servers were completely connected with each other. We were also able to switch from servers to servers if the connectivity was successful. Overall, I think that this was a good introduction to the course and to the following discussions and activities.

"I affirm that I will not give or receive any unauthorized help on this activity and that all work will be my own."