Name: Alexis Neil D. Deniega	**Date Performed:** August 8th, 2025
Course/Section: CPE31S4	**Date Submitted:** August 8th, 2025
Instructor: Engr. Robin Valenzuela	**Semester and SY:** SY 2025-26 1st Sem

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

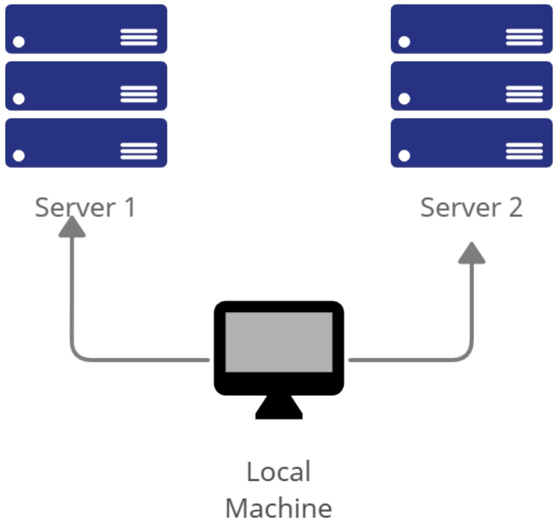
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

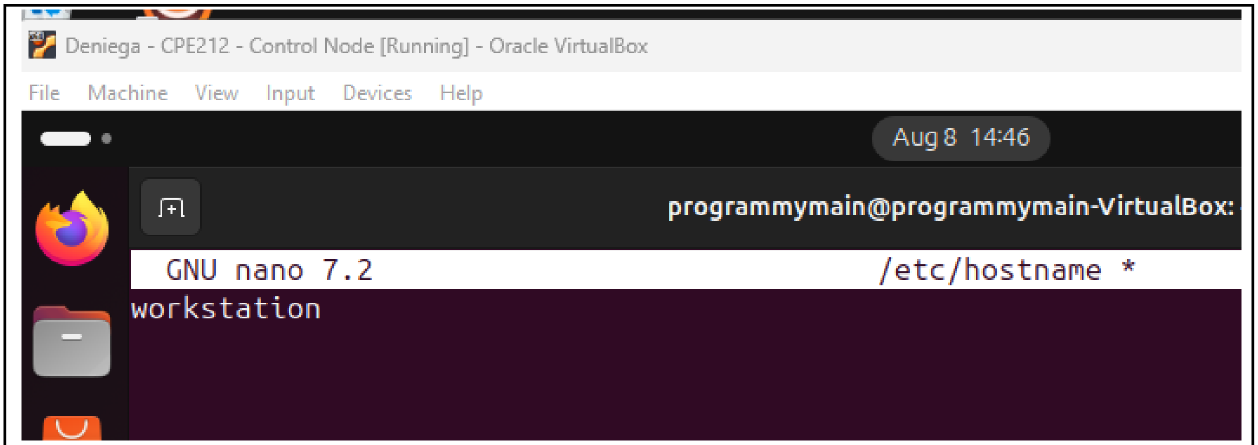
Network Topology:

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*).

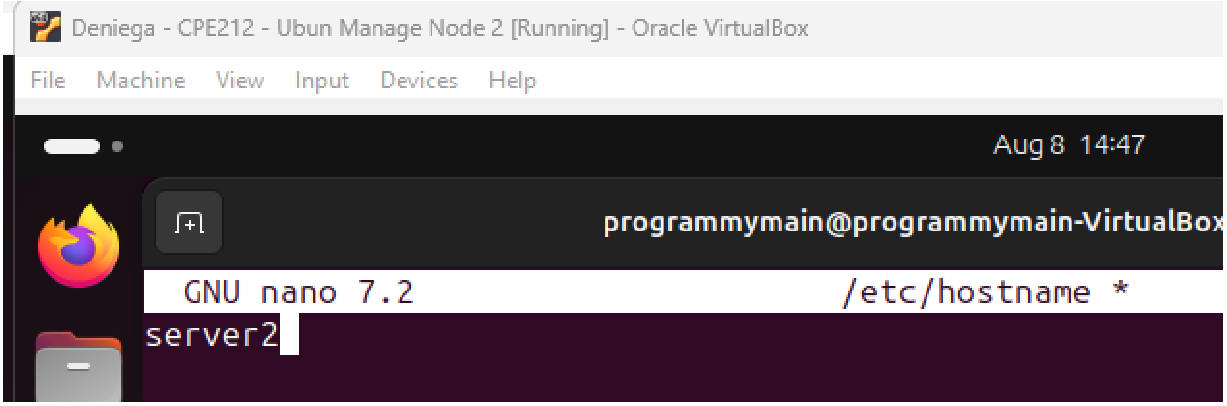


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

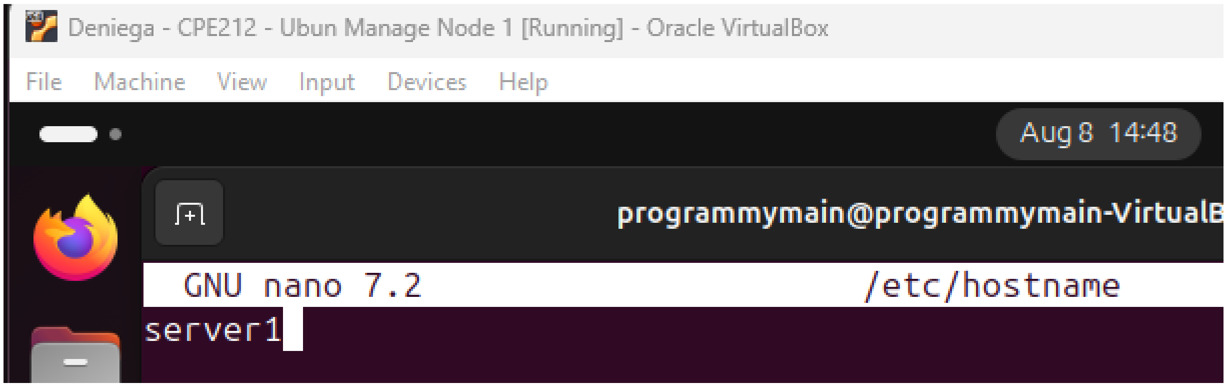
1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1
   1.2 Use server2 for Server 2
   1.3 Use workstation for the Local Machine


Deniega - CPE212 - Control Node [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:46
programmymain@programmymain-VirtualBox:

```
  GNU nano 7.2                          /etc/hostname *
workstation
```

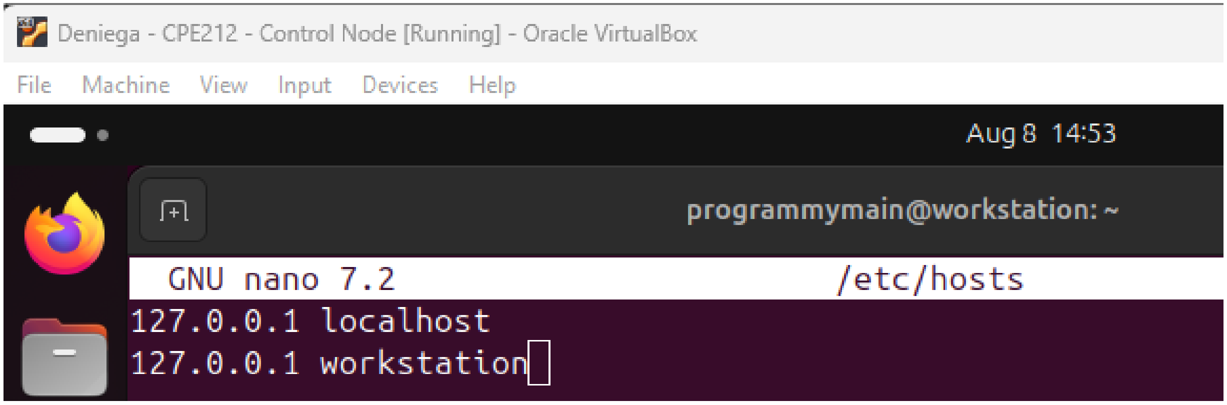

Deniega - CPE212 - Ubun Manage Node 2 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:47
programmymain@programmymain-VirtualBox

```
  GNU nano 7.2                          /etc/hostname *
server2
```

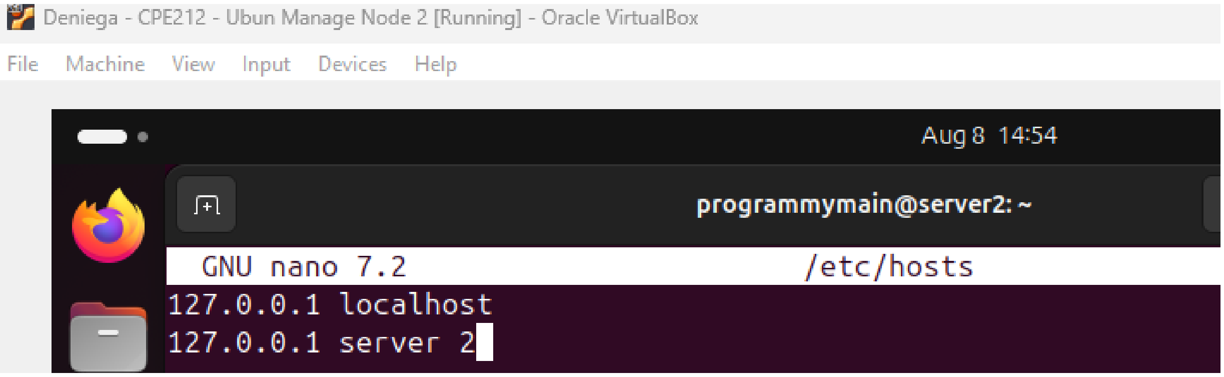

Deniega - CPE212 - Ubun Manage Node 1 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:48
programmymain@programmymain-VirtualB

```
  GNU nano 7.2                          /etc/hostname
server1
```

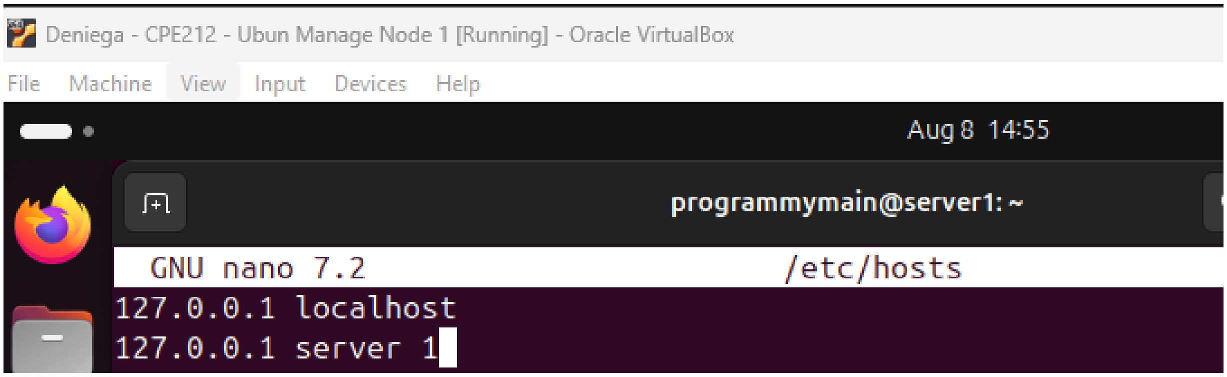
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
    2.1 Type 127.0.0.1 server 1 for Server 1
    2.2 Type 127.0.0.1 server 2 for Server 2
    2.3 Type 127.0.0.1 workstation for the Local Machine


Deniega - CPE212 - Control Node [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:53
programmymain@workstation: ~

```
  GNU nano 7.2                          /etc/hosts
127.0.0.1 localhost
127.0.0.1 workstation
```


Deniega - CPE212 - Ubun Manage Node 2 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:54
programmymain@server2: ~

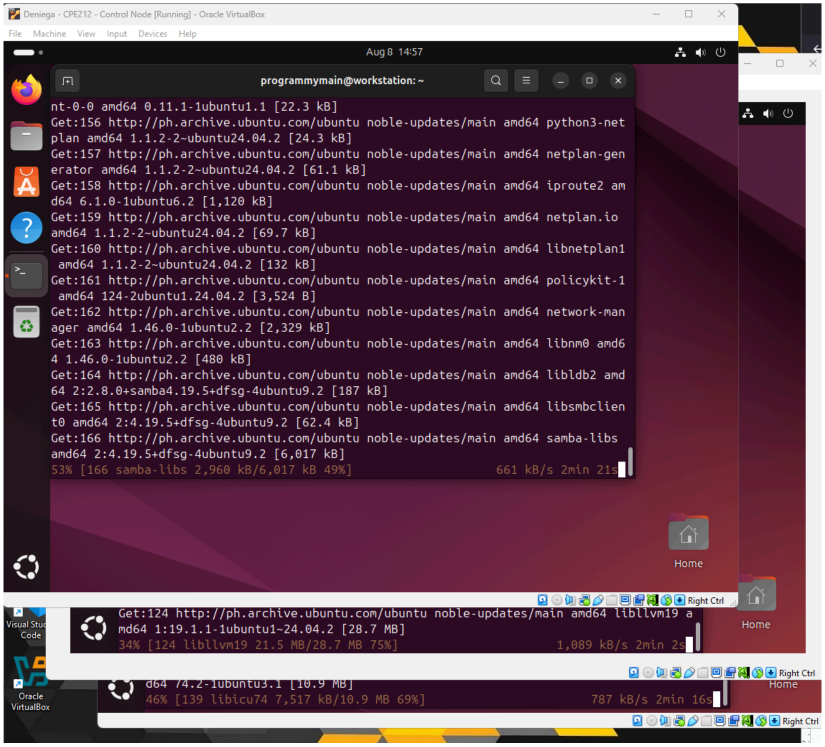
```
  GNU nano 7.2                          /etc/hosts
127.0.0.1 localhost
127.0.0.1 server 2
```


Deniega - CPE212 - Ubun Manage Node 1 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:55
programmymain@server1: ~

```
  GNU nano 7.2                          /etc/hosts
127.0.0.1 localhost
127.0.0.1 server 1
```

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

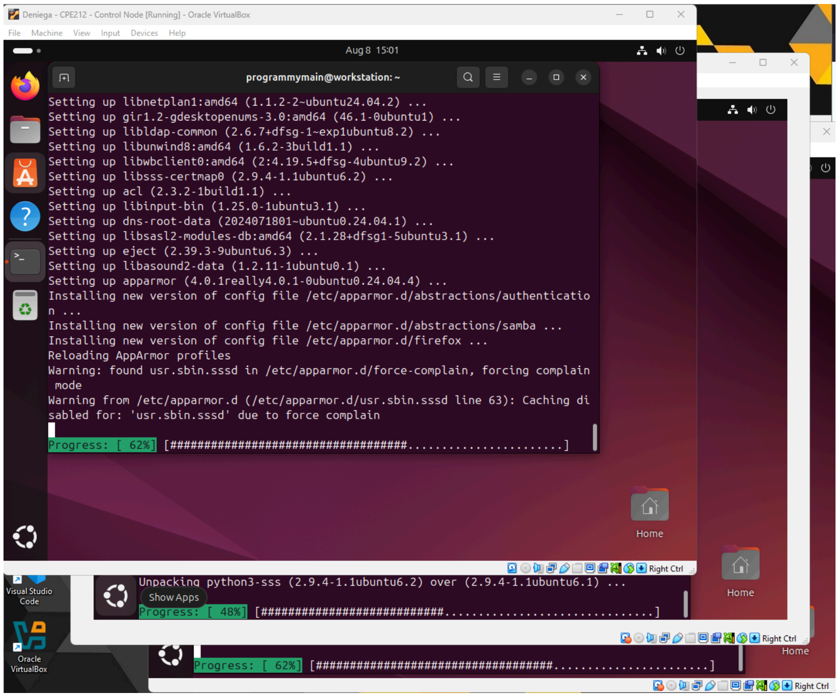
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.


Deniega - CPE212 - Control Node [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 14:57
programmymain@workstation: ~
Home
Right Ctrl
Visual Studio Code
Oracle VirtualBox

```
nt-0-0 amd64 0.11.1-1ubuntu1.1 [22.3 kB]
Get:156 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-net
plan amd64 1.1.2-2~ubuntu24.04.2 [24.3 kB]
Get:157 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 netplan-gen
erator amd64 1.1.2-2~ubuntu24.04.2 [61.1 kB]
Get:158 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 iproute2 am
d64 6.1.0-1ubuntu6.2 [1,120 kB]
Get:159 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 netplan.io
amd64 1.1.2-2~ubuntu24.04.2 [69.7 kB]
Get:160 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libnetplan1
 amd64 1.1.2-2~ubuntu24.04.2 [132 kB]
Get:161 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 policykit-1
 amd64 124-2ubuntu1.24.04.2 [3,524 B]
Get:162 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 network-man
ager amd64 1.46.0-1ubuntu2.2 [2,329 kB]
Get:163 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libnm0 amd6
4 1.46.0-1ubuntu2.2 [480 kB]
Get:164 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libldb2 amd
64 2:2.8.0+samba4.19.5+dfsg-4ubuntu9.2 [187 kB]
Get:165 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libsmbclien
t0 amd64 2:4.19.5+dfsg-4ubuntu9.2 [62.4 kB]
Get:166 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 samba-libs
amd64 2:4.19.5+dfsg-4ubuntu9.2 [6,017 kB]
53% [166 samba-libs 2,960 kB/6,017 kB 49%]                 661 kB/s 2min 21s
```

```
Get:124 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libllvm19 a
md64 1:19.1.1-1ubuntu1~24.04.2 [28.7 MB]
34% [124 libllvm19 21.5 MB/28.7 MB 75%]                 1,089 kB/s 2min 2s
```

```
d64 74.2-1ubuntu3.1 [10.9 MB]
46% [139 libicu74 7,517 kB/10.9 MB 69%]                 787 kB/s 2min 16s
```

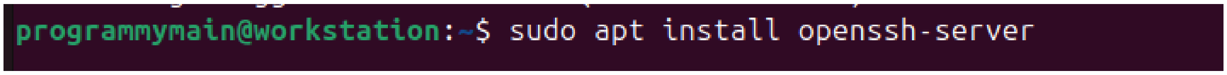

Deniega - CPE212 - Control Node [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 8 15:01
programmymain@workstation: ~
Home
Right Ctrl
Show Apps
Visual Studio Code
Oracle VirtualBox

```
Setting up libnetplan1:amd64 (1.1.2-2~ubuntu24.04.2) ...
Setting up gir1.2-gdesktopenums-3.0:amd64 (46.1-0ubuntu1) ...
Setting up libldap-common (2.6.7+dfsg-1~exp1ubuntu8.2) ...
Setting up libunwind8:amd64 (1.6.2-3build1.1) ...
Setting up libwbclient0:amd64 (2:4.19.5+dfsg-4ubuntu9.2) ...
Setting up libsss-certmap0 (2.9.4-1.1ubuntu6.2) ...
Setting up acl (2.3.2-1build1.1) ...
Setting up libinput-bin (1.25.0-1ubuntu3.1) ...
Setting up dns-root-data (2024071801~ubuntu0.24.04.1) ...
Setting up libsasl2-modules-db:amd64 (2.1.28+dfsg1-5ubuntu3.1) ...
Setting up eject (2.39.3-9ubuntu6.3) ...
Setting up libasound2-data (1.2.11-1ubuntu0.1) ...
Setting up apparmor (4.0.1really4.0.1-0ubuntu0.24.04.4) ...
Installing new version of config file /etc/apparmor.d/abstractions/authenticatio
n ...
Installing new version of config file /etc/apparmor.d/abstractions/samba ...
Installing new version of config file /etc/apparmor.d/firefox ...
Reloading AppArmor profiles
Warning: found usr.sbin.sssd in /etc/apparmor.d/force-complain, forcing complain
 mode
Warning from /etc/apparmor.d (/etc/apparmor.d/usr.sbin.sssd line 63): Caching di
sabled for: 'usr.sbin.sssd' due to force complain

Progress: [ 62%] [####################################......................]
```

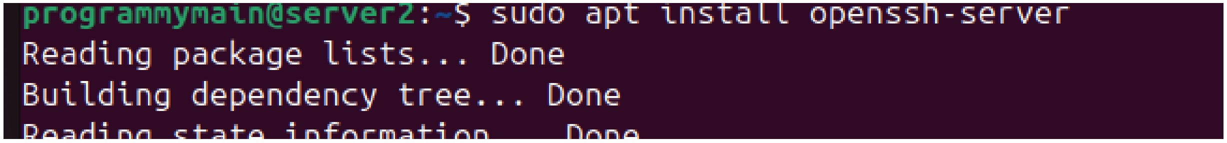
```
Unpacking python3-sss (2.9.4-1.1ubuntu6.2) over (2.9.4-1.1ubuntu6.1) ...
Progress: [ 48%] [############################..............................]
```

```
Progress: [ 62%] [####################################......................]
```

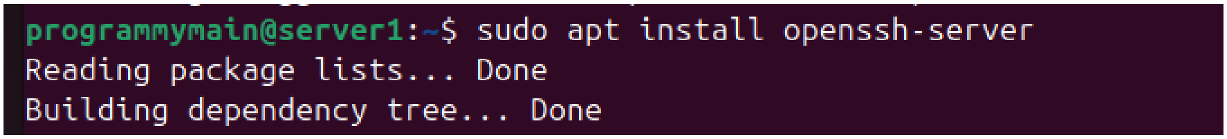
2. Install the SSH server using the command *sudo apt install openssh-server*.



```
programmymain@workstation:~$ sudo apt install openssh-server
```

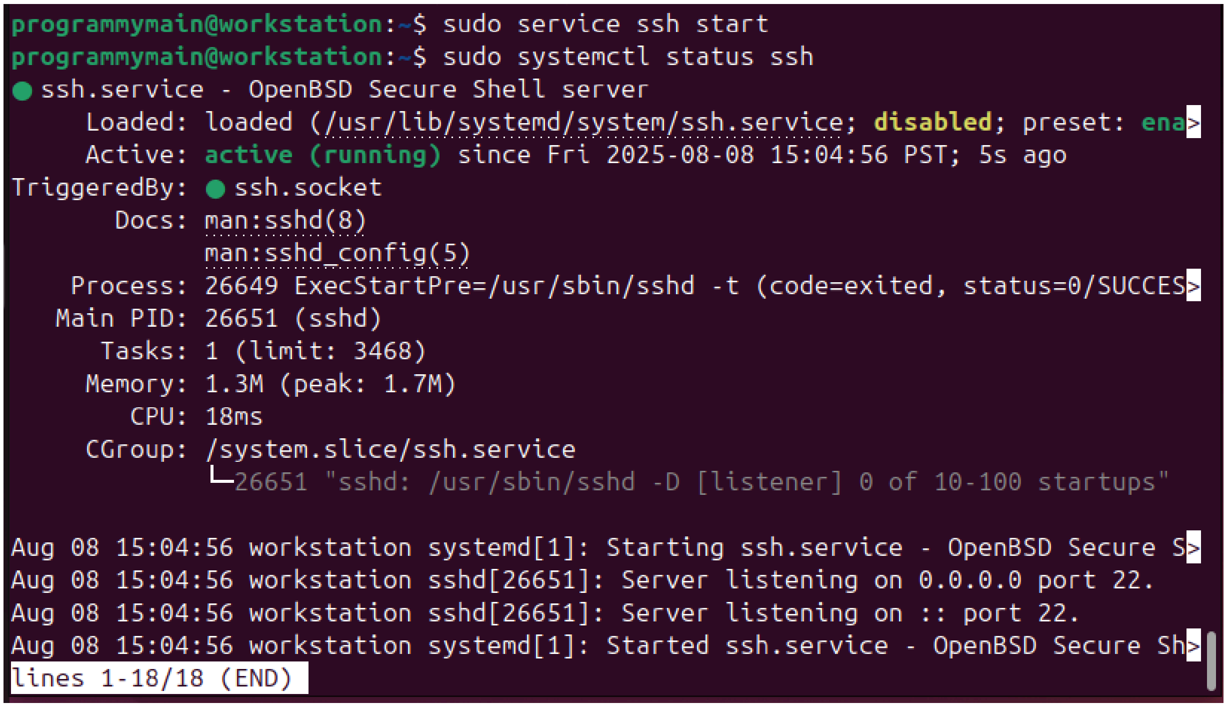


```
programmymain@server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

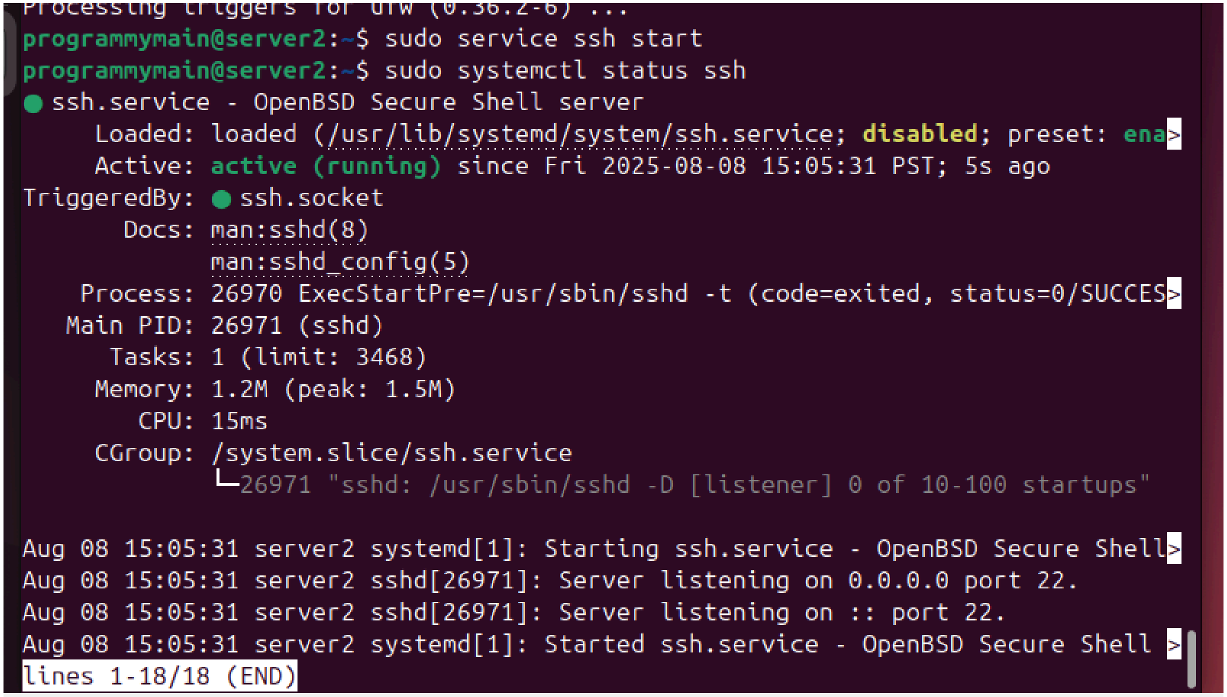


```
programmymain@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
```

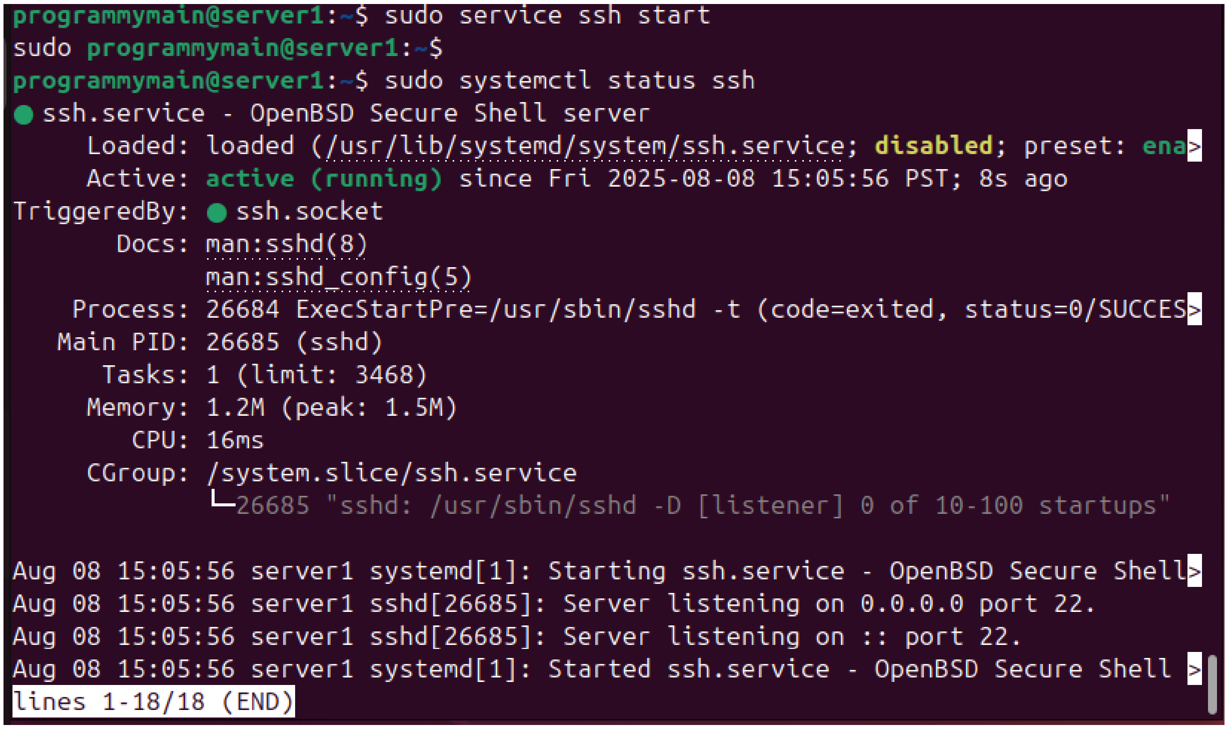
3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*



```
programmymain@workstation:~$ sudo service ssh start
programmymain@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2025-08-08 15:04:56 PST; 5s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 26649 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 26651 (sshd)
      Tasks: 1 (limit: 3468)
     Memory: 1.3M (peak: 1.7M)
        CPU: 18ms
     CGroup: /system.slice/ssh.service
             └─26651 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 08 15:04:56 workstation systemd[1]: Starting ssh.service - OpenBSD Secure S>
Aug 08 15:04:56 workstation sshd[26651]: Server listening on 0.0.0.0 port 22.
Aug 08 15:04:56 workstation sshd[26651]: Server listening on :: port 22.
Aug 08 15:04:56 workstation systemd[1]: Started ssh.service - OpenBSD Secure Sh>
lines 1-18/18 (END)
```

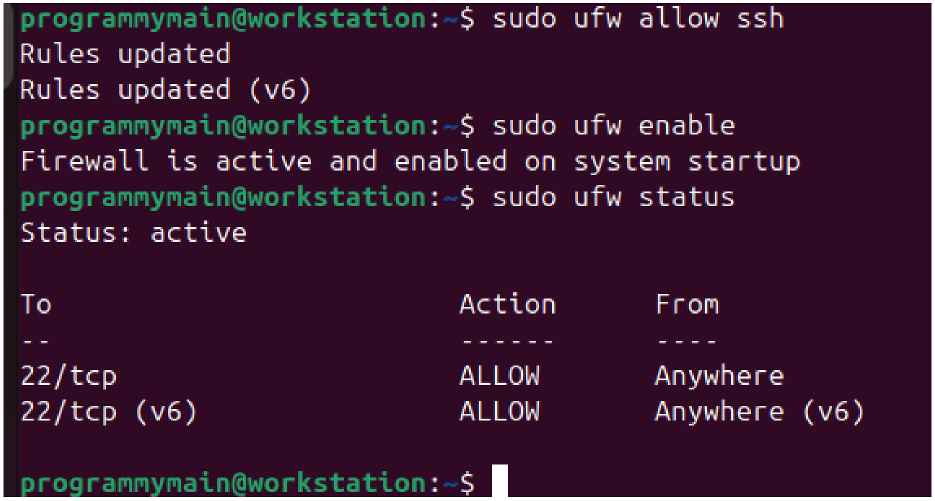


```
Processing triggers for ufw (0.36.2-6) ...
programmymain@server2:~$ sudo service ssh start
programmymain@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2025-08-08 15:05:31 PST; 5s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 26970 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 26971 (sshd)
      Tasks: 1 (limit: 3468)
     Memory: 1.2M (peak: 1.5M)
        CPU: 15ms
     CGroup: /system.slice/ssh.service
             └─26971 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 08 15:05:31 server2 systemd[1]: Starting ssh.service - OpenBSD Secure Shell>
Aug 08 15:05:31 server2 sshd[26971]: Server listening on 0.0.0.0 port 22.
Aug 08 15:05:31 server2 sshd[26971]: Server listening on :: port 22.
Aug 08 15:05:31 server2 systemd[1]: Started ssh.service - OpenBSD Secure Shell >
lines 1-18/18 (END)
```

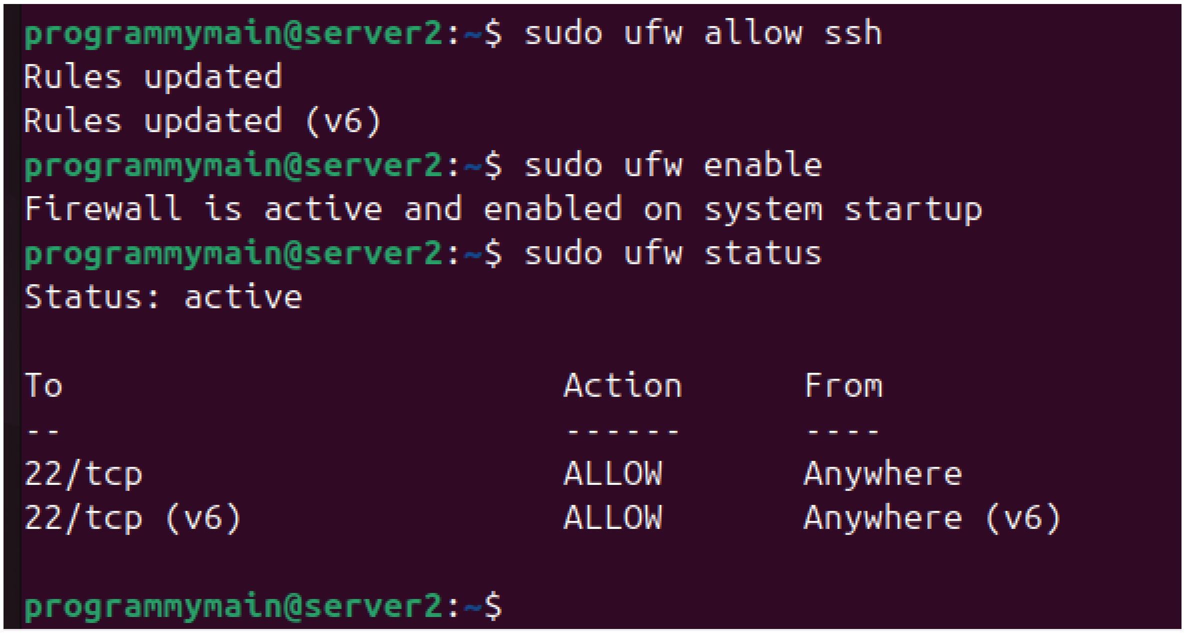


```
programmymain@server1:~$ sudo service ssh start
sudo programmymain@server1:~$
programmymain@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2025-08-08 15:05:56 PST; 8s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 26684 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 26685 (sshd)
      Tasks: 1 (limit: 3468)
     Memory: 1.2M (peak: 1.5M)
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─26685 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 08 15:05:56 server1 systemd[1]: Starting ssh.service - OpenBSD Secure Shell>
Aug 08 15:05:56 server1 sshd[26685]: Server listening on 0.0.0.0 port 22.
Aug 08 15:05:56 server1 sshd[26685]: Server listening on :: port 22.
Aug 08 15:05:56 server1 systemd[1]: Started ssh.service - OpenBSD Secure Shell >
lines 1-18/18 (END)
```

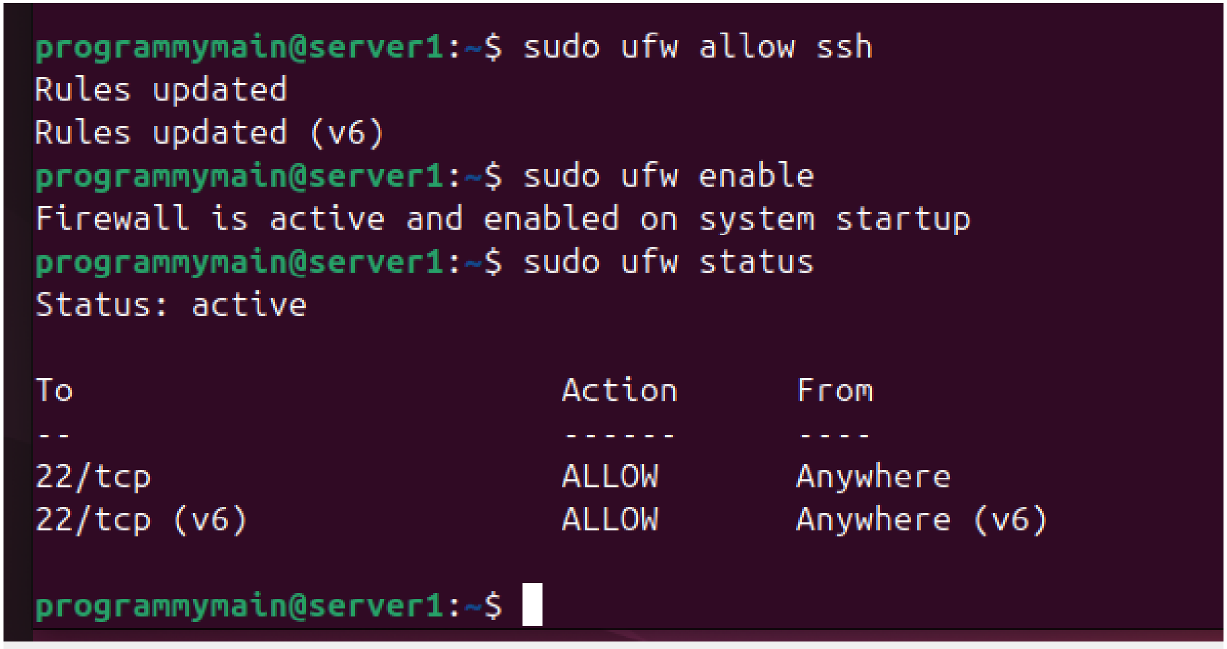
4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*



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

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

programmymain@workstation:~$ 
```



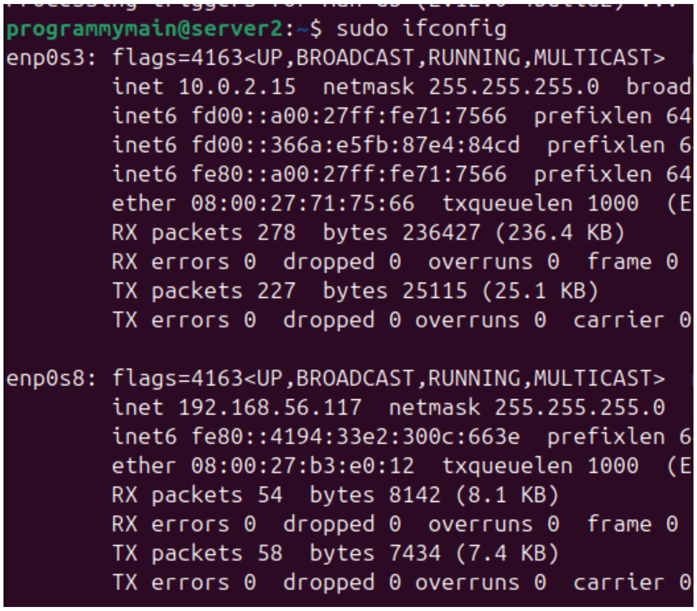
```
programmymain@server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
programmymain@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
programmymain@server2:~$ sudo ufw status
Status: active

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

programmymain@server2:~$
```



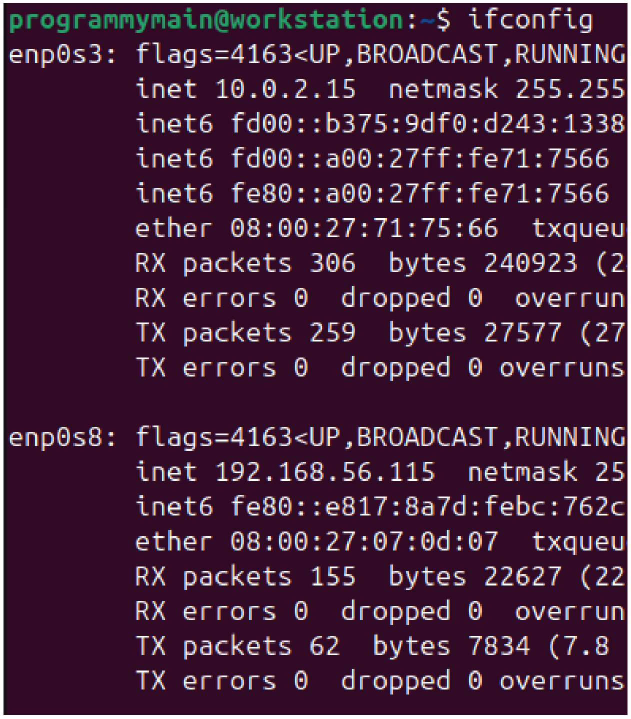
```
programmymain@server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
programmymain@server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
programmymain@server1:~$ sudo ufw status
Status: active

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

programmymain@server1:~$ 
```

Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

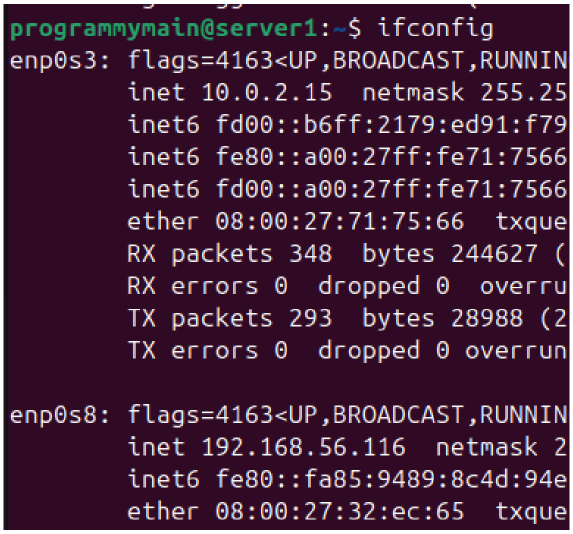
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.
   1.1 Server 1 IP address: 192.168.56.**116**
   1.2 Server 2 IP address: 192.168.56.**117**
   1.3 Workstation IP address: 192.168.56.**115**



```
programmymain@server2:~$ sudo ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
        inet 10.0.2.15  netmask 255.255.255.0  broad
        inet6 fd00::a00:27ff:fe71:7566  prefixlen 64
        inet6 fd00::366a:e5fb:87e4:84cd  prefixlen 6
        inet6 fe80::a00:27ff:fe71:7566  prefixlen 64
        ether 08:00:27:71:75:66  txqueuelen 1000  (E
        RX packets 278  bytes 236427 (236.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 227  bytes 25115 (25.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
        inet 192.168.56.117  netmask 255.255.255.0
        inet6 fe80::4194:33e2:300c:663e  prefixlen 6
        ether 08:00:27:b3:e0:12  txqueuelen 1000  (E
        RX packets 54  bytes 8142 (8.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 58  bytes 7434 (7.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0
```



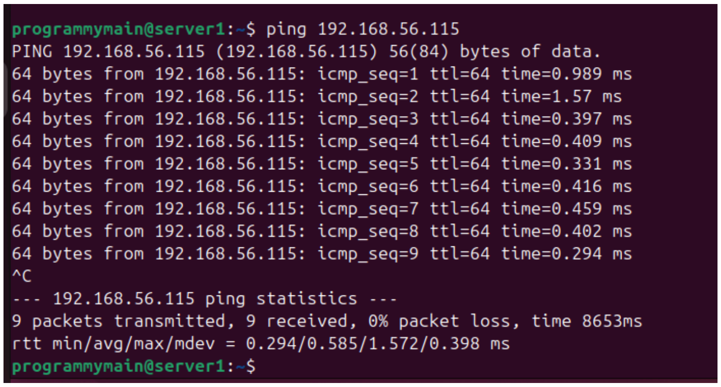
```
programmymain@workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING
        inet 10.0.2.15  netmask 255.255
        inet6 fd00::b375:9df0:d243:1338
        inet6 fd00::a00:27ff:fe71:7566
        inet6 fe80::a00:27ff:fe71:7566
        ether 08:00:27:71:75:66  txqueu
        RX packets 306  bytes 240923 (2
        RX errors 0  dropped 0  overrun
        TX packets 259  bytes 27577 (27
        TX errors 0  dropped 0 overruns

enp0s8: flags=4163<UP,BROADCAST,RUNNING
        inet 192.168.56.115  netmask 25
        inet6 fe80::e817:8a7d:febc:762c
        ether 08:00:27:07:0d:07  txqueu
        RX packets 155  bytes 22627 (22
        RX errors 0  dropped 0  overrun
        TX packets 62  bytes 7834 (7.8
        TX errors 0  dropped 0 overruns
```



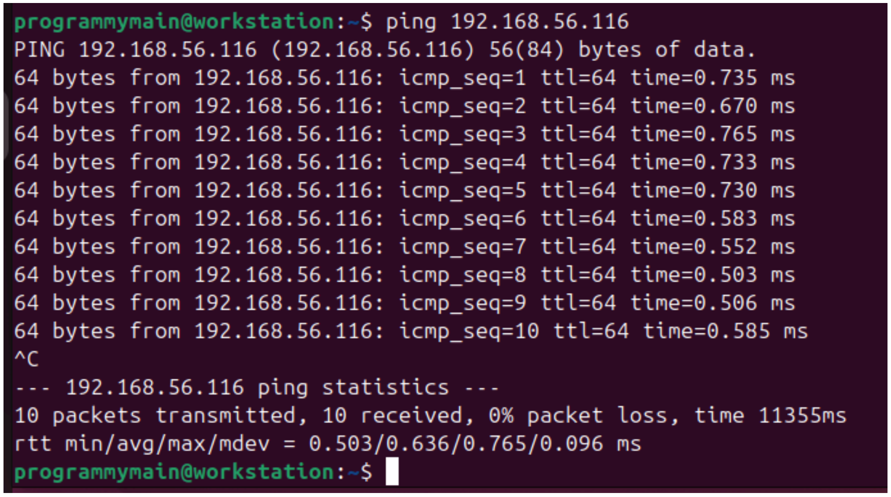
```
programmymain@server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNIN
        inet 10.0.2.15  netmask 255.25
        inet6 fd00::b6ff:2179:ed91:f79
        inet6 fe80::a00:27ff:fe71:7566
        inet6 fd00::a00:27ff:fe71:7566
        ether 08:00:27:71:75:66  txque
        RX packets 348  bytes 244627 (
        RX errors 0  dropped 0  overru
        TX packets 293  bytes 28988 (2
        TX errors 0  dropped 0 overrun

enp0s8: flags=4163<UP,BROADCAST,RUNNIN
        inet 192.168.56.116  netmask 2
        inet6 fe80::fa85:9489:8c4d:94e
        ether 08:00:27:32:ec:65  txque
```

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

Workstation to Server 1 (and v.v.)

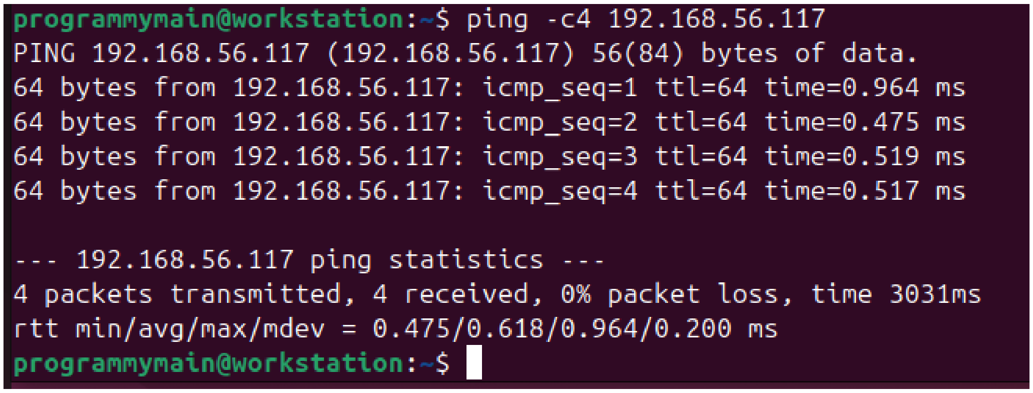


```
programmymain@server1:~$ ping 192.168.56.115
PING 192.168.56.115 (192.168.56.115) 56(84) bytes of data.
64 bytes from 192.168.56.115: icmp_seq=1 ttl=64 time=0.989 ms
64 bytes from 192.168.56.115: icmp_seq=2 ttl=64 time=1.57 ms
64 bytes from 192.168.56.115: icmp_seq=3 ttl=64 time=0.397 ms
64 bytes from 192.168.56.115: icmp_seq=4 ttl=64 time=0.409 ms
64 bytes from 192.168.56.115: icmp_seq=5 ttl=64 time=0.331 ms
64 bytes from 192.168.56.115: icmp_seq=6 ttl=64 time=0.416 ms
64 bytes from 192.168.56.115: icmp_seq=7 ttl=64 time=0.459 ms
64 bytes from 192.168.56.115: icmp_seq=8 ttl=64 time=0.402 ms
64 bytes from 192.168.56.115: icmp_seq=9 ttl=64 time=0.294 ms
^C
--- 192.168.56.115 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8653ms
rtt min/avg/max/mdev = 0.294/0.585/1.572/0.398 ms
programmymain@server1:~$
```

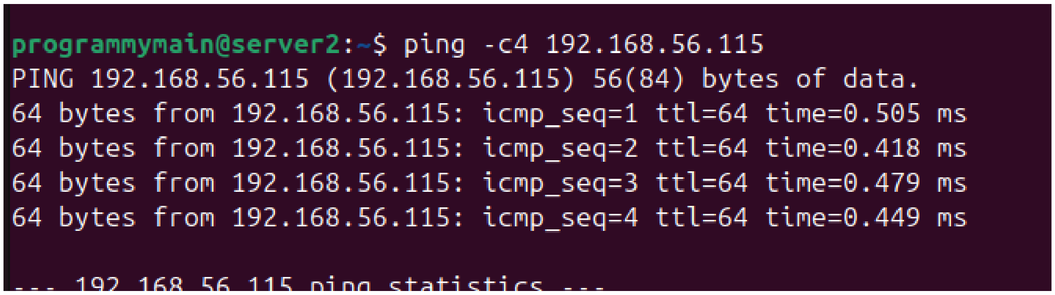


```
programmymain@workstation:~$ ping 192.168.56.116
PING 192.168.56.116 (192.168.56.116) 56(84) bytes of data.
64 bytes from 192.168.56.116: icmp_seq=1 ttl=64 time=0.735 ms
64 bytes from 192.168.56.116: icmp_seq=2 ttl=64 time=0.670 ms
64 bytes from 192.168.56.116: icmp_seq=3 ttl=64 time=0.765 ms
64 bytes from 192.168.56.116: icmp_seq=4 ttl=64 time=0.733 ms
64 bytes from 192.168.56.116: icmp_seq=5 ttl=64 time=0.730 ms
64 bytes from 192.168.56.116: icmp_seq=6 ttl=64 time=0.583 ms
64 bytes from 192.168.56.116: icmp_seq=7 ttl=64 time=0.552 ms
64 bytes from 192.168.56.116: icmp_seq=8 ttl=64 time=0.503 ms
64 bytes from 192.168.56.116: icmp_seq=9 ttl=64 time=0.506 ms
64 bytes from 192.168.56.116: icmp_seq=10 ttl=64 time=0.585 ms
^C
--- 192.168.56.116 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 11355ms
rtt min/avg/max/mdev = 0.503/0.636/0.765/0.096 ms
programmymain@workstation:~$
```

Workstation to Server 2 (and v.v.)

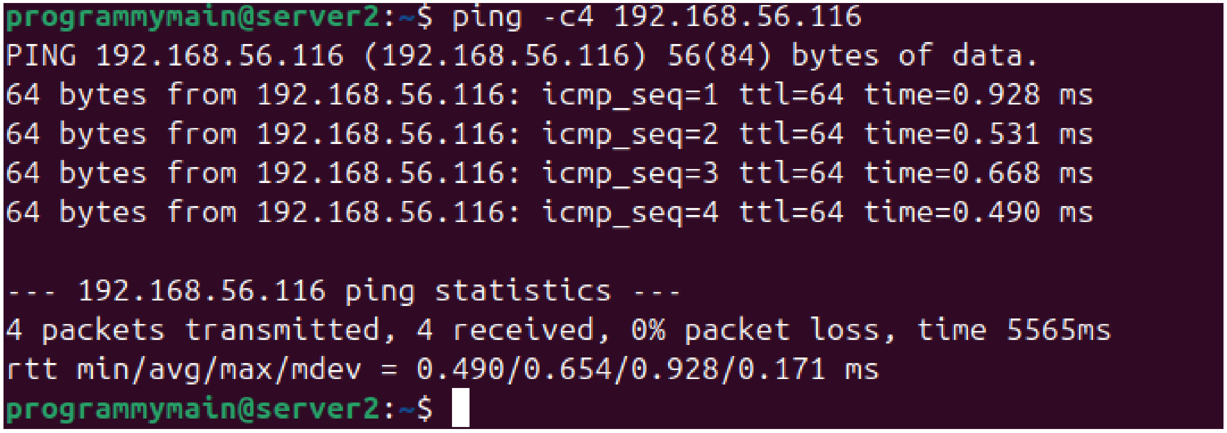


```
programmymain@workstation:~$ ping -c4 192.168.56.117
PING 192.168.56.117 (192.168.56.117) 56(84) bytes of data.
64 bytes from 192.168.56.117: icmp_seq=1 ttl=64 time=0.964 ms
64 bytes from 192.168.56.117: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 192.168.56.117: icmp_seq=3 ttl=64 time=0.519 ms
64 bytes from 192.168.56.117: icmp_seq=4 ttl=64 time=0.517 ms

--- 192.168.56.117 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3031ms
rtt min/avg/max/mdev = 0.475/0.618/0.964/0.200 ms
programmymain@workstation:~$
```

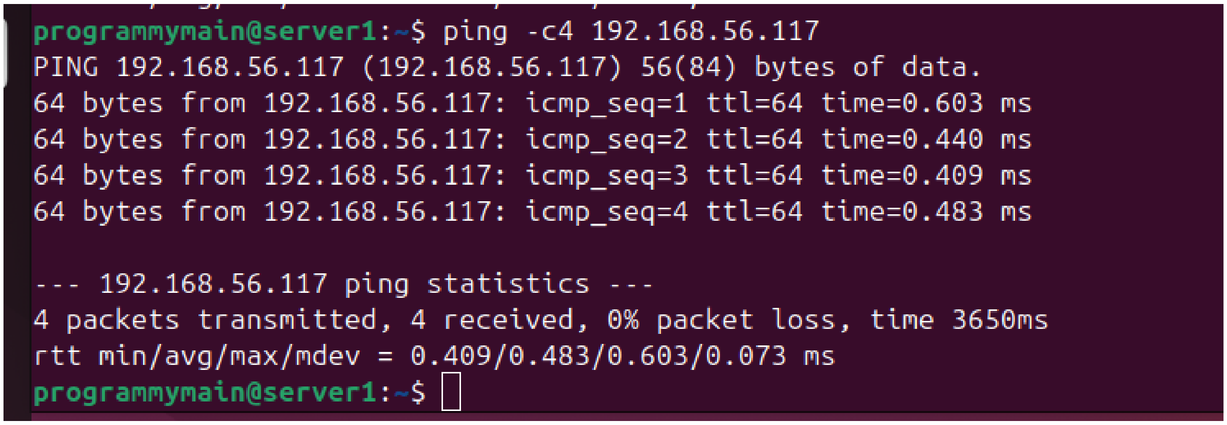


```
programmymain@server2:~$ ping -c4 192.168.56.115
PING 192.168.56.115 (192.168.56.115) 56(84) bytes of data.
64 bytes from 192.168.56.115: icmp_seq=1 ttl=64 time=0.505 ms
64 bytes from 192.168.56.115: icmp_seq=2 ttl=64 time=0.418 ms
64 bytes from 192.168.56.115: icmp_seq=3 ttl=64 time=0.479 ms
64 bytes from 192.168.56.115: icmp_seq=4 ttl=64 time=0.449 ms

--- 192.168.56.115 ping statistics ---
```

Server 1 to 2 and vice versa



```
programmymain@server2:~$ ping -c4 192.168.56.116
PING 192.168.56.116 (192.168.56.116) 56(84) bytes of data.
64 bytes from 192.168.56.116: icmp_seq=1 ttl=64 time=0.928 ms
64 bytes from 192.168.56.116: icmp_seq=2 ttl=64 time=0.531 ms
64 bytes from 192.168.56.116: icmp_seq=3 ttl=64 time=0.668 ms
64 bytes from 192.168.56.116: icmp_seq=4 ttl=64 time=0.490 ms

--- 192.168.56.116 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 5565ms
rtt min/avg/max/mdev = 0.490/0.654/0.928/0.171 ms
programmymain@server2:~$ 
```



```
programmymain@server1:~$ ping -c4 192.168.56.117
PING 192.168.56.117 (192.168.56.117) 56(84) bytes of data.
64 bytes from 192.168.56.117: icmp_seq=1 ttl=64 time=0.603 ms
64 bytes from 192.168.56.117: icmp_seq=2 ttl=64 time=0.440 ms
64 bytes from 192.168.56.117: icmp_seq=3 ttl=64 time=0.409 ms
64 bytes from 192.168.56.117: icmp_seq=4 ttl=64 time=0.483 ms

--- 192.168.56.117 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3650ms
rtt min/avg/max/mdev = 0.409/0.483/0.603/0.073 ms
programmymain@server1:~$ 
```

Server 1 IP address: 192.168.56.**116**
Server 2 IP address: 192.168.56.**117**
Workstation IP address: 192.168.56.**115**

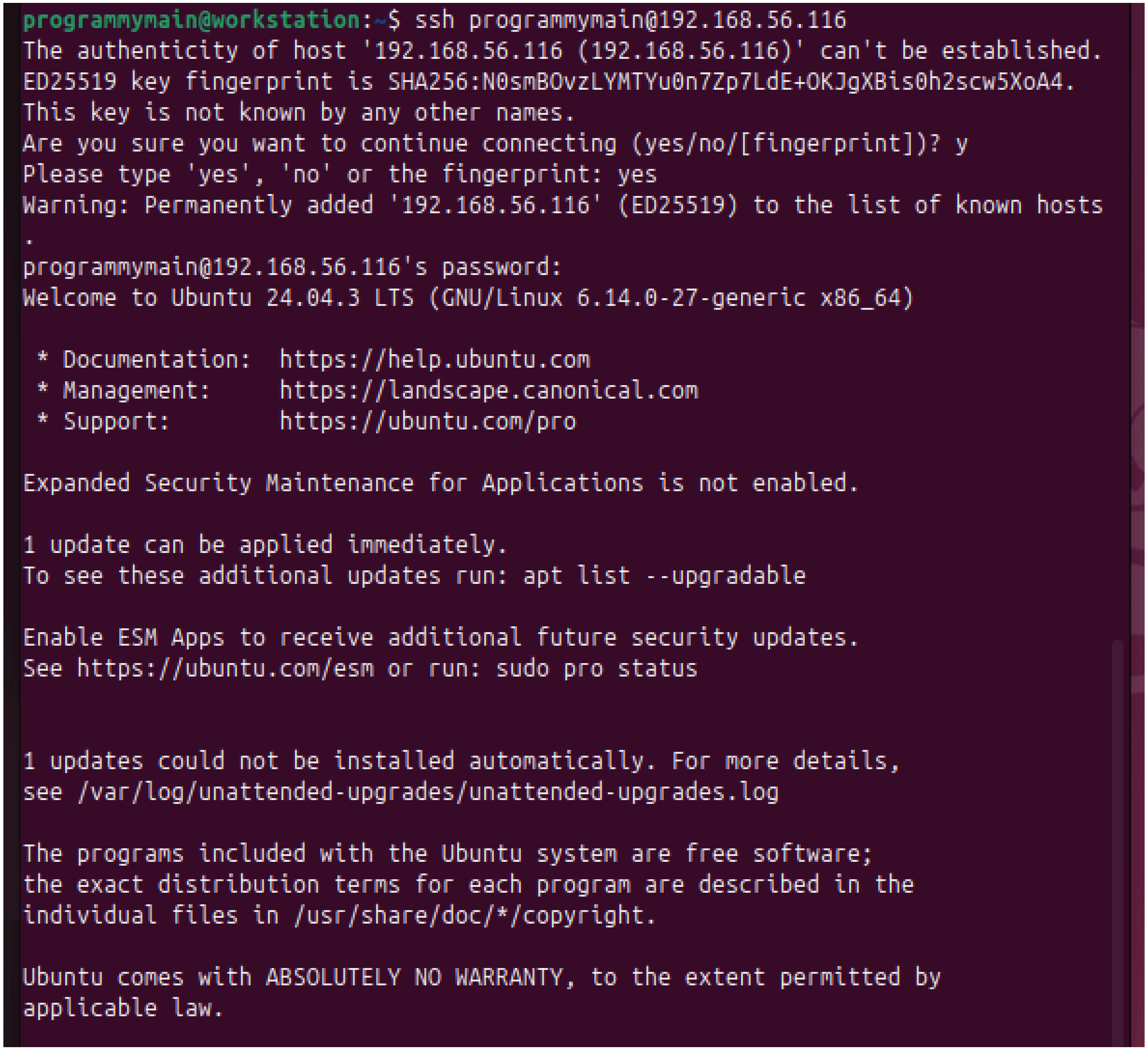
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

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

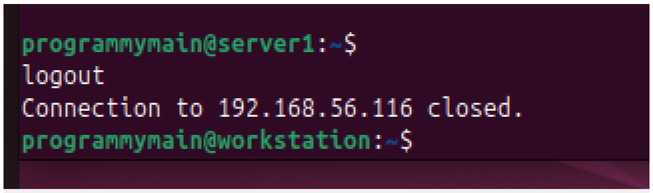
1.2 Enter the password for server 1 when prompted

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



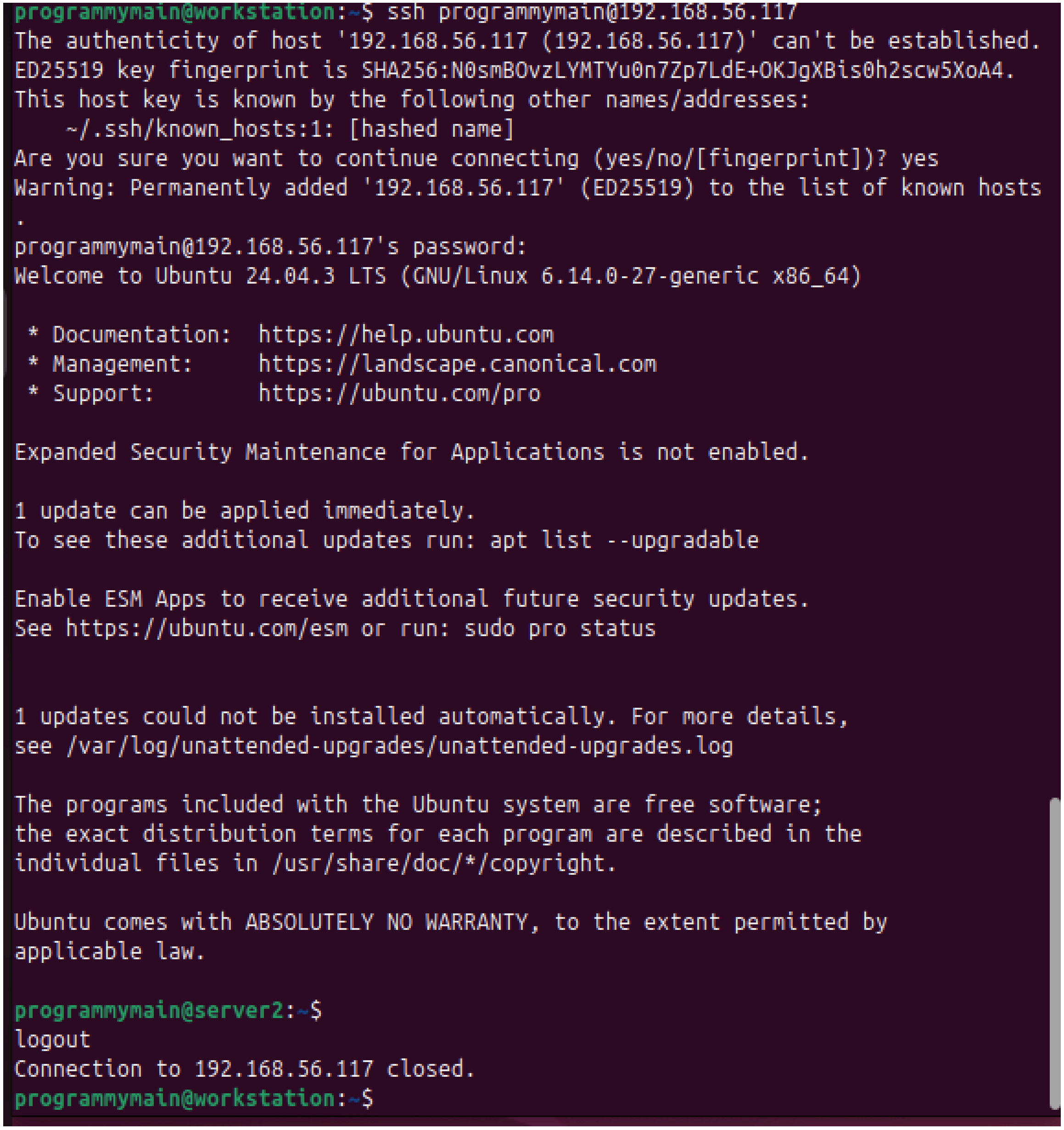
```
programmymain@workstation:~$ ssh programmymain@192.168.56.116
The authenticity of host '192.168.56.116 (192.168.56.116)' can't be established.
ED25519 key fingerprint is SHA256:N0smBOvzLYMTYu0n7Zp7LdE+OKJgXBis0h2scw5XoA4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.116' (ED25519) to the list of known hosts
.
programmymain@192.168.56.116's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

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.
```

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



```
programmymain@server1:~$
logout
Connection to 192.168.56.116 closed.
programmymain@workstation:~$
```

3. Do the same for Server 2.



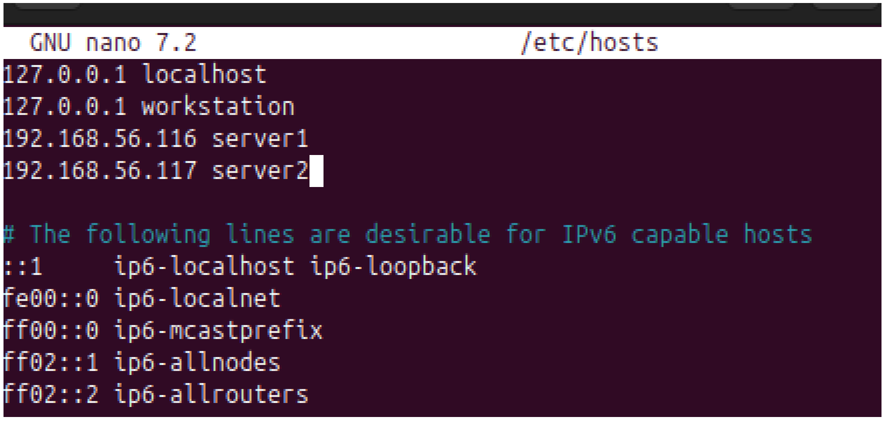
```
programmymain@workstation:~$ ssh programmymain@192.168.56.117
The authenticity of host '192.168.56.117 (192.168.56.117)' can't be established.
ED25519 key fingerprint is SHA256:N0smBOvzLYMTYu0n7Zp7LdE+OKJgXBis0h2scw5XoA4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.117' (ED25519) to the list of known hosts
.
programmymain@192.168.56.117's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

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.

programmymain@server2:~$
logout
Connection to 192.168.56.117 closed.
programmymain@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)

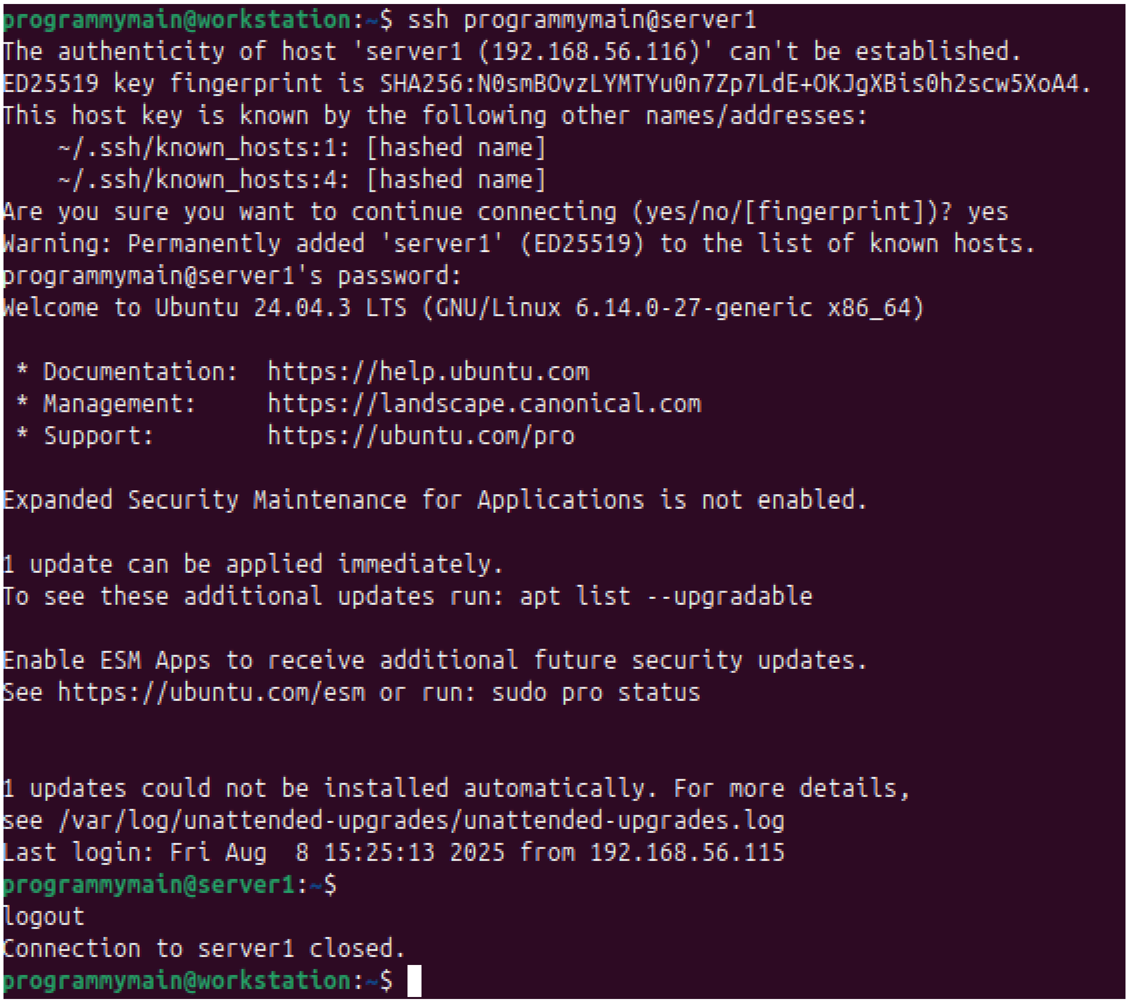
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

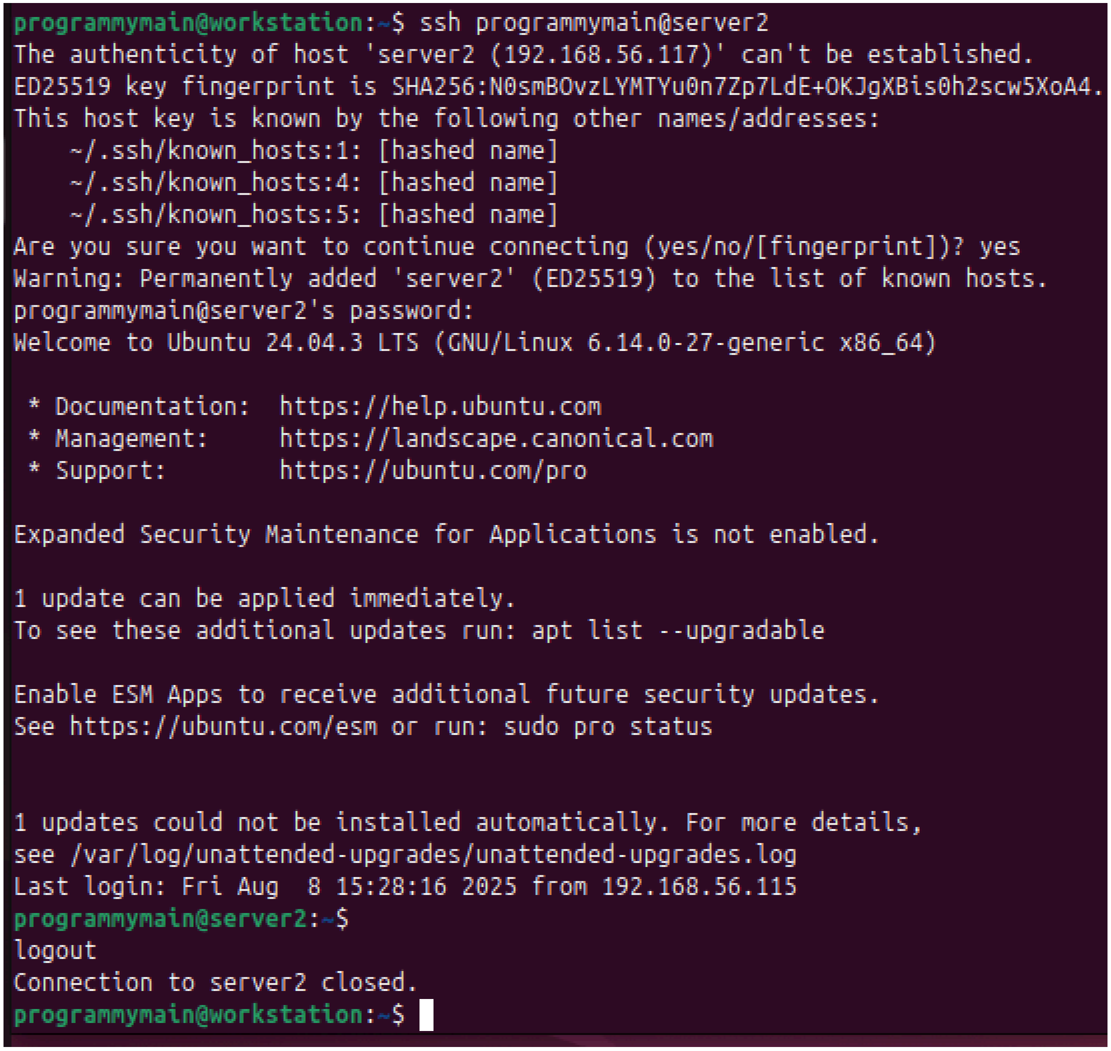


```
  GNU nano 7.2                          /etc/hosts
127.0.0.1 localhost
127.0.0.1 workstation
192.168.56.116 server1
192.168.56.117 server2

# 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.



```
programmymain@workstation:~$ ssh programmymain@server1
The authenticity of host 'server1 (192.168.56.116)' can't be established.
ED25519 key fingerprint is SHA256:N0smBOvzLYMTYu0n7Zp7LdE+OKJgXBis0h2scw5XoA4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server1' (ED25519) to the list of known hosts.
programmymain@server1's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Fri Aug  8 15:25:13 2025 from 192.168.56.115
programmymain@server1:~$
logout
Connection to server1 closed.
programmymain@workstation:~$
```



```
programmymain@workstation:~$ ssh programmymain@server2
The authenticity of host 'server2 (192.168.56.117)' can't be established.
ED25519 key fingerprint is SHA256:N0smBOvzLYMTYu0n7Zp7LdE+OKJgXBis0h2scw5XoA4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2' (ED25519) to the list of known hosts.
programmymain@server2's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Fri Aug  8 15:28:16 2025 from 192.168.56.115
programmymain@server2:~$
logout
Connection to server2 closed.
programmymain@workstation:~$ 
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   - *When I put the IP address and its corresponding hostname on the /etc/hosts file, the two new lines act as some sort of a table, where the hosts are the rows, and the IP-name combo as the columns. When we SSH the hostname, the program searches for matches, and replaces the hostname with its corresponding IP address, then does whatever SSH does afterwards.*

2. How secured is SSH?
   - *Each device has a private key that is used in tandem with the public key, and they are often big and long, like SHA256, to prevent any brute-force guessing. When mixed, the two communicators are the only ones capable of deciphering*

it, even if it catches out in the public, since they have the private key hashes to themselves and only themselves, making it secure and safe to use.