Name: De Omampo, Julius Mark A.	**Date Performed:** 08/25/2024
Course/Section: CPE212 – CPE31S2	**Date Submitted:** 08/25/2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem. (2024-2025)

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

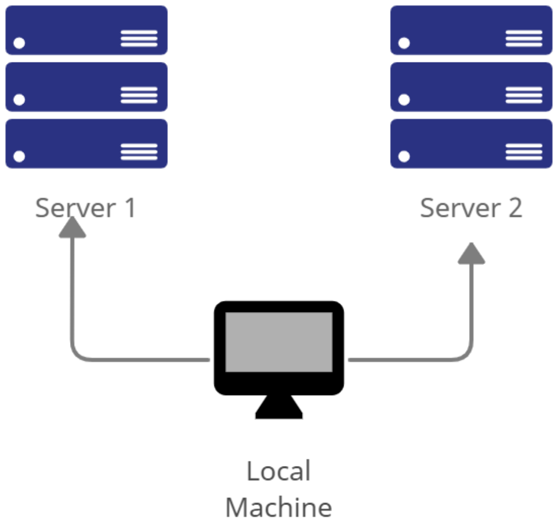
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

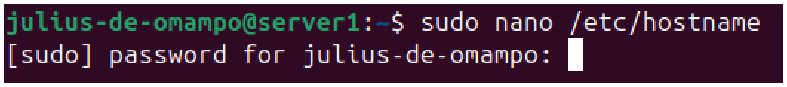
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



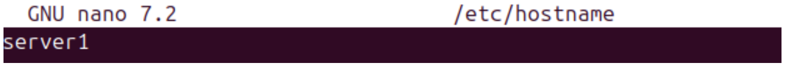
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

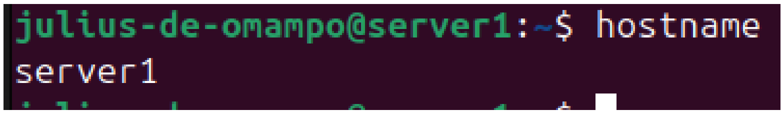
1.1 Use server1 for Server 1



```
julius-de-omampo@server1:~$ sudo nano /etc/hostname
[sudo] password for julius-de-omampo:
```

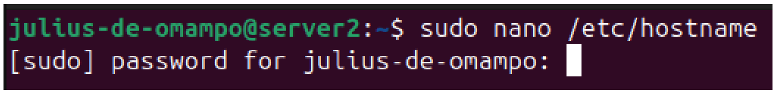


```
  GNU nano 7.2                              /etc/hostname
server1
```

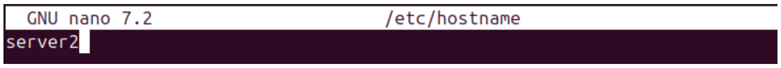


```
julius-de-omampo@server1:~$ hostname
server1
```

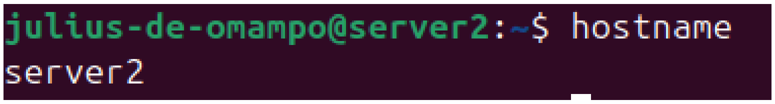
1.2 Use server2 for Server 2



```
julius-de-omampo@server2:~$ sudo nano /etc/hostname
[sudo] password for julius-de-omampo:
```

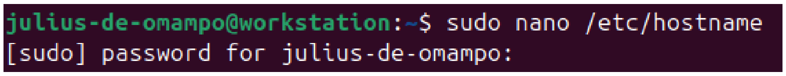


```
  GNU nano 7.2                          /etc/hostname
server2
```

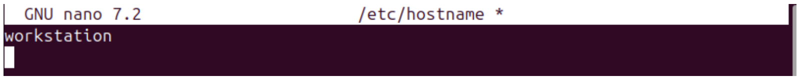


```
julius-de-omampo@server2:~$ hostname
server2
```

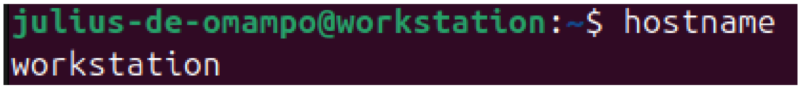
1.3 Use workstation for the Local Machine



```
julius-de-omampo@workstation:~$ sudo nano /etc/hostname
[sudo] password for julius-de-omampo:
```



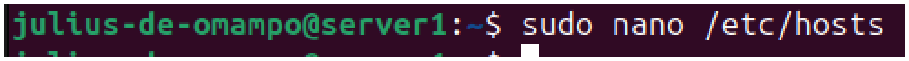
```
  GNU nano 7.2                          /etc/hostname *
workstation
```



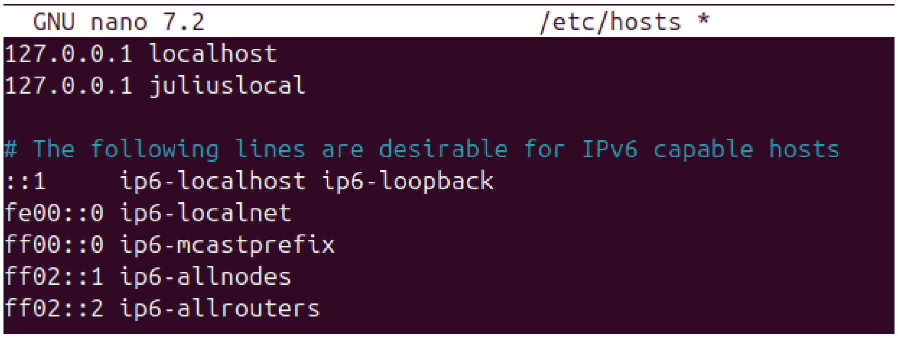
```
julius-de-omampo@workstation:~$ hostname
workstation
```

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

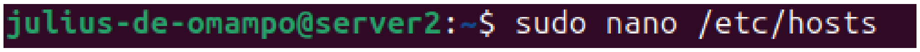


```
julius-de-omampo@server1:~$ sudo nano /etc/hosts
```

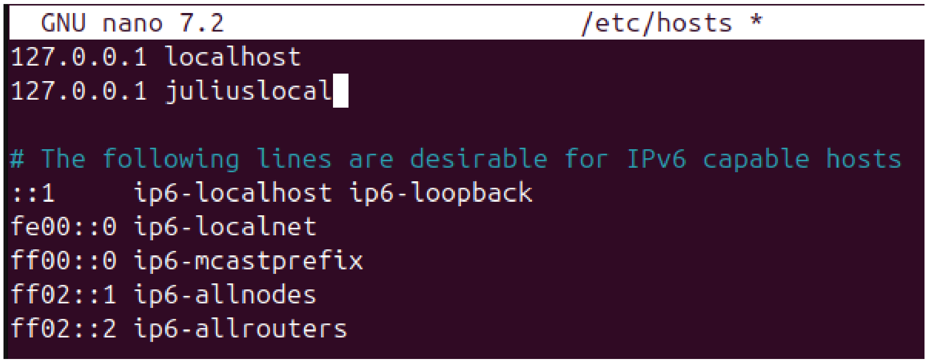


```
  GNU nano 7.2                          /etc/hosts *
127.0.0.1 localhost
127.0.0.1 juliuslocal

# 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
```

2.2 Type 127.0.0.1 server 2 for Server 2

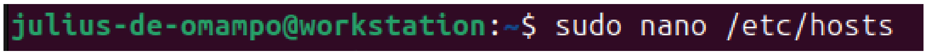


```
julius-de-omampo@server2:~$ sudo nano /etc/hosts
```

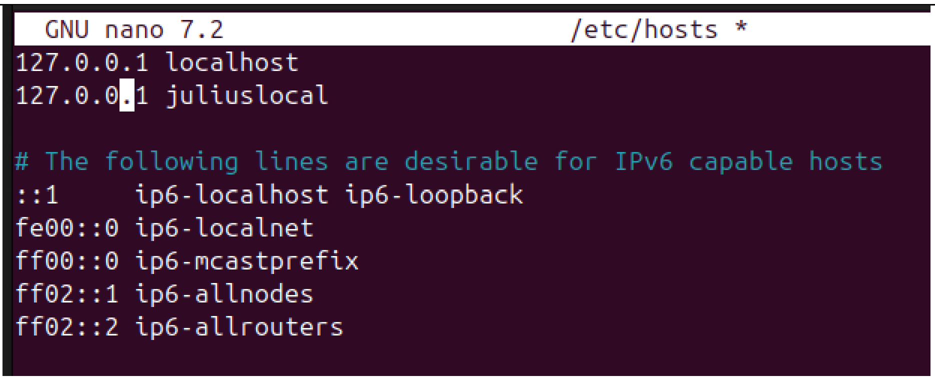


```
  GNU nano 7.2                          /etc/hosts *
127.0.0.1 localhost
127.0.0.1 juliuslocal

# 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
```

2.3 Type 127.0.0.1 workstation for the Local Machine



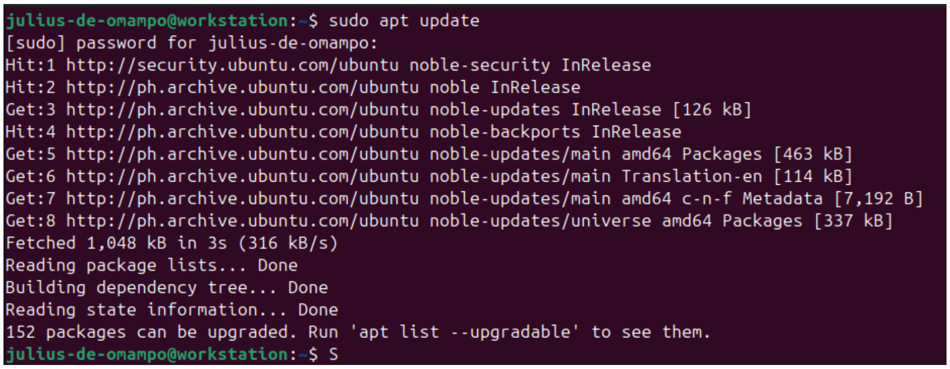
```
julius-de-omampo@workstation:~$ sudo nano /etc/hosts
```



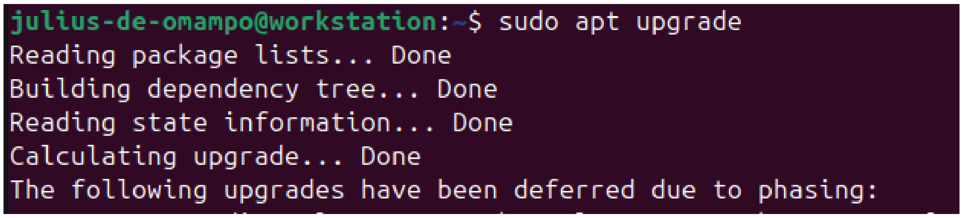
```
  GNU nano 7.2                          /etc/hosts *
127.0.0.1 localhost
127.0.0.1 juliuslocal

# 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
```

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

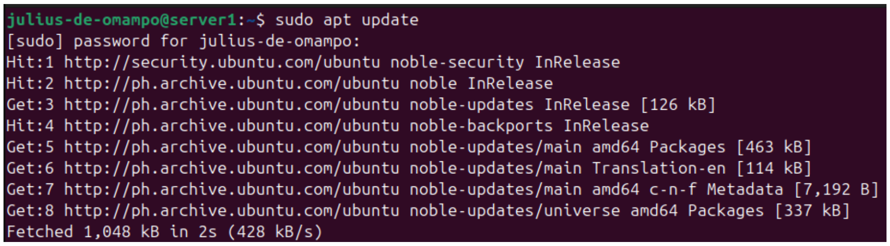
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.



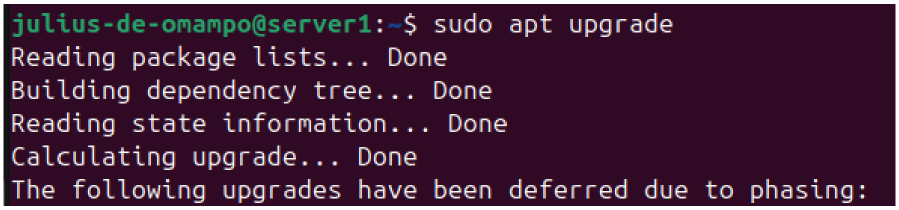
```
julius-de-omampo@workstation:~$ sudo apt update
[sudo] password for julius-de-omampo:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Get:6 http://ph.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [114 kB]
Get:7 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7,192 B]
Get:8 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Fetched 1,048 kB in 3s (316 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
152 packages can be upgraded. Run 'apt list --upgradable' to see them.
julius-de-omampo@workstation:~$ S
```



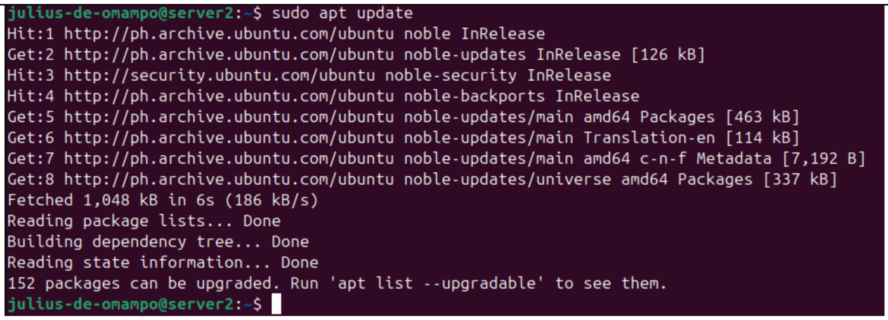
```
julius-de-omampo@workstation:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
```



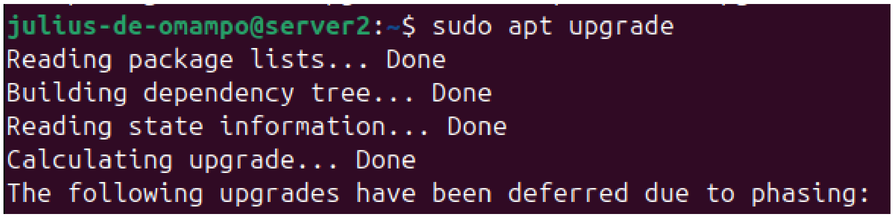
```
julius-de-omampo@server1:~$ sudo apt update
[sudo] password for julius-de-omampo:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Get:6 http://ph.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [114 kB]
Get:7 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7,192 B]
Get:8 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Fetched 1,048 kB in 2s (428 kB/s)
```



```
julius-de-omampo@server1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
```

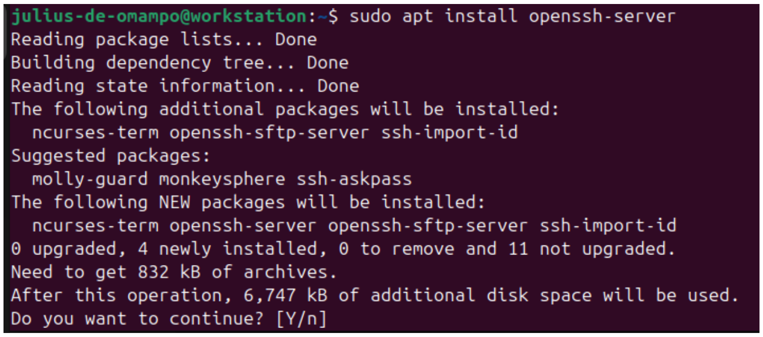


```
julius-de-omampo@server2:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Get:6 http://ph.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [114 kB]
Get:7 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7,192 B]
Get:8 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Fetched 1,048 kB in 6s (186 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
152 packages can be upgraded. Run 'apt list --upgradable' to see them.
julius-de-omampo@server2:~$
```

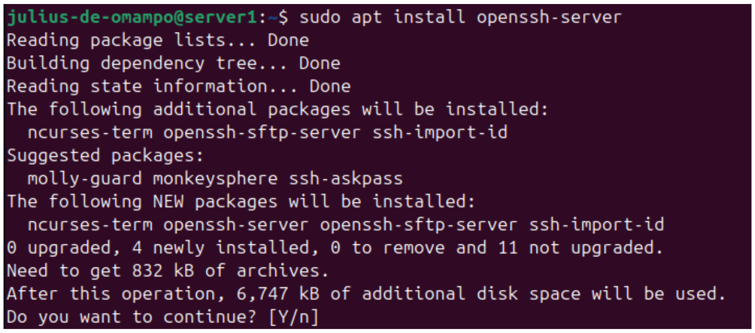


```
julius-de-omampo@server2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
```

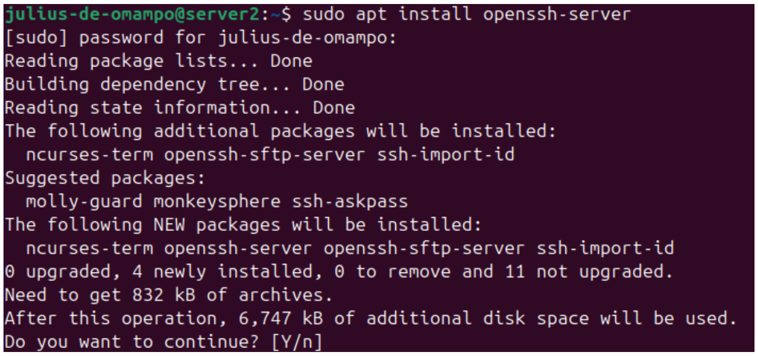
2. Install the SSH server using the command *sudo apt install openssh-server*.



```
julius-de-omampo@workstation:~$ 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 11 not upgraded.
Need to get 832 kB of archives.
After this operation, 6,747 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

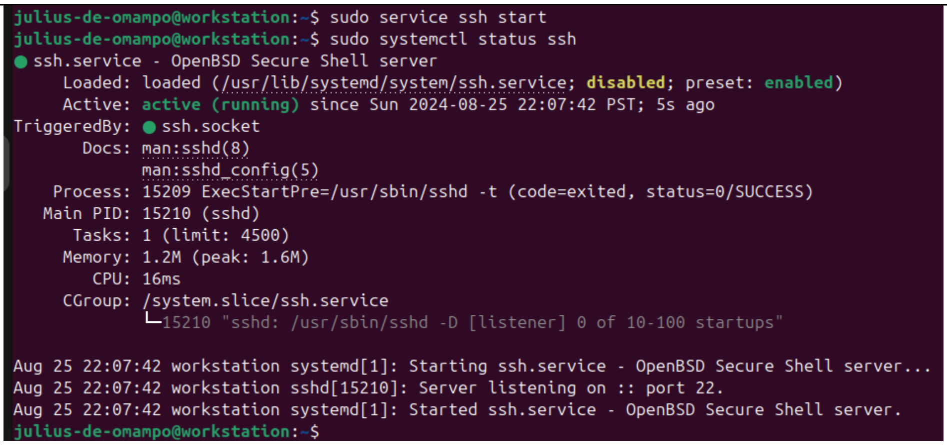


```
julius-de-omampo@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere 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 11 not upgraded.
Need to get 832 kB of archives.
After this operation, 6,747 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

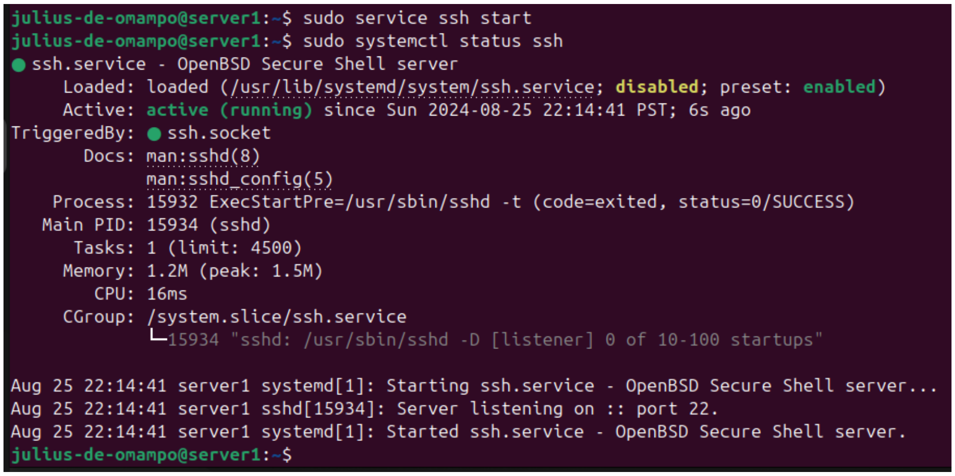


```
julius-de-omampo@server2:~$ sudo apt install openssh-server
[sudo] password for julius-de-omampo:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
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 11 not upgraded.
Need to get 832 kB of archives.
After this operation, 6,747 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

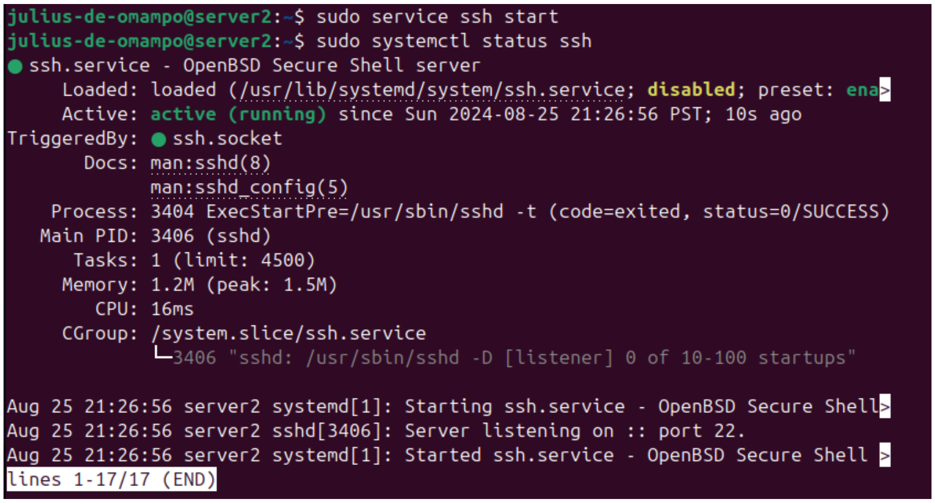
3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*



```
julius-de-omampo@workstation:~$ sudo service ssh start
julius-de-omampo@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Sun 2024-08-25 22:07:42 PST; 5s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 15209 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 15210 (sshd)
      Tasks: 1 (limit: 4500)
     Memory: 1.2M (peak: 1.6M)
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─15210 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 22:07:42 workstation systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Aug 25 22:07:42 workstation sshd[15210]: Server listening on :: port 22.
Aug 25 22:07:42 workstation systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
julius-de-omampo@workstation:~$
```

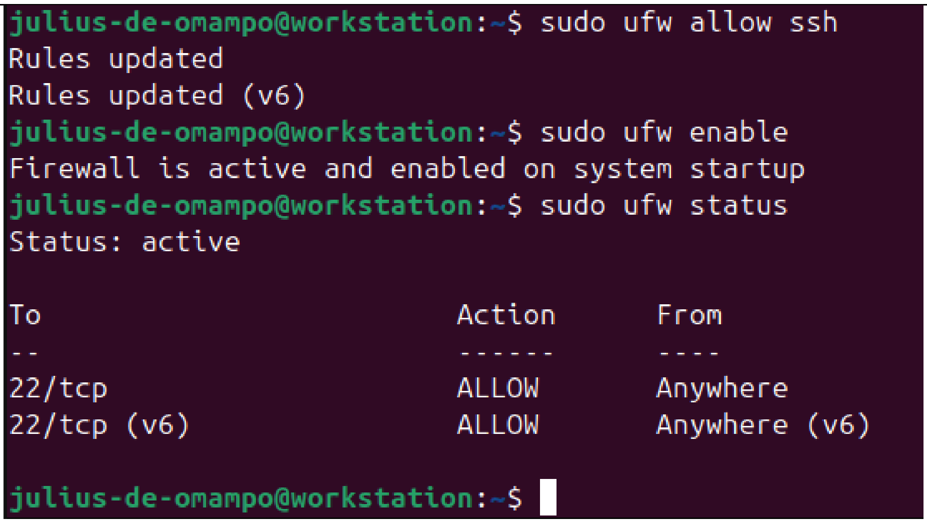


```
julius-de-omampo@server1:~$ sudo service ssh start
julius-de-omampo@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Sun 2024-08-25 22:14:41 PST; 6s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 15932 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 15934 (sshd)
      Tasks: 1 (limit: 4500)
     Memory: 1.2M (peak: 1.5M)
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─15934 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 22:14:41 server1 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Aug 25 22:14:41 server1 sshd[15934]: Server listening on :: port 22.
Aug 25 22:14:41 server1 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
julius-de-omampo@server1:~$
```

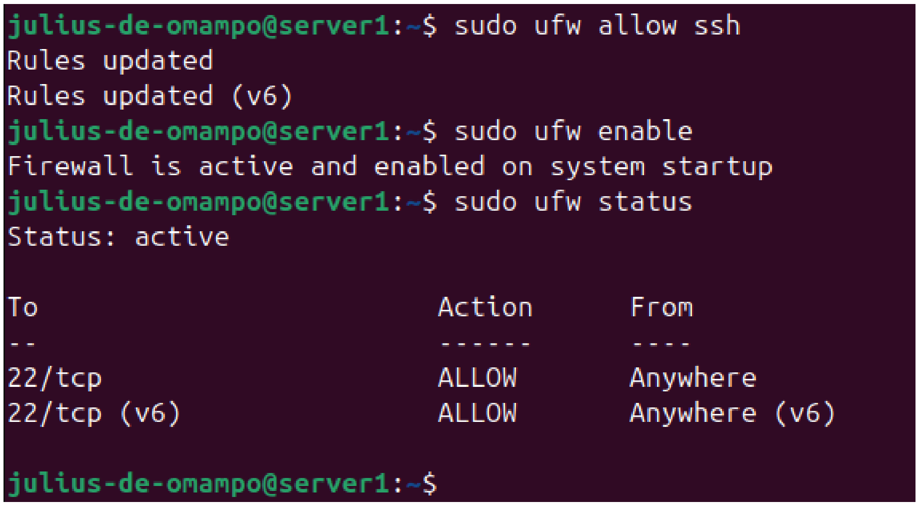


```
julius-de-omampo@server2:~$ sudo service ssh start
julius-de-omampo@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Sun 2024-08-25 21:26:56 PST; 10s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3404 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3406 (sshd)
      Tasks: 1 (limit: 4500)
     Memory: 1.2M (peak: 1.5M)
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─3406 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 21:26:56 server2 systemd[1]: Starting ssh.service - OpenBSD Secure Shell>
Aug 25 21:26:56 server2 sshd[3406]: Server listening on :: port 22.
Aug 25 21:26:56 server2 systemd[1]: Started ssh.service - OpenBSD Secure Shell >
lines 1-17/17 (END)
```

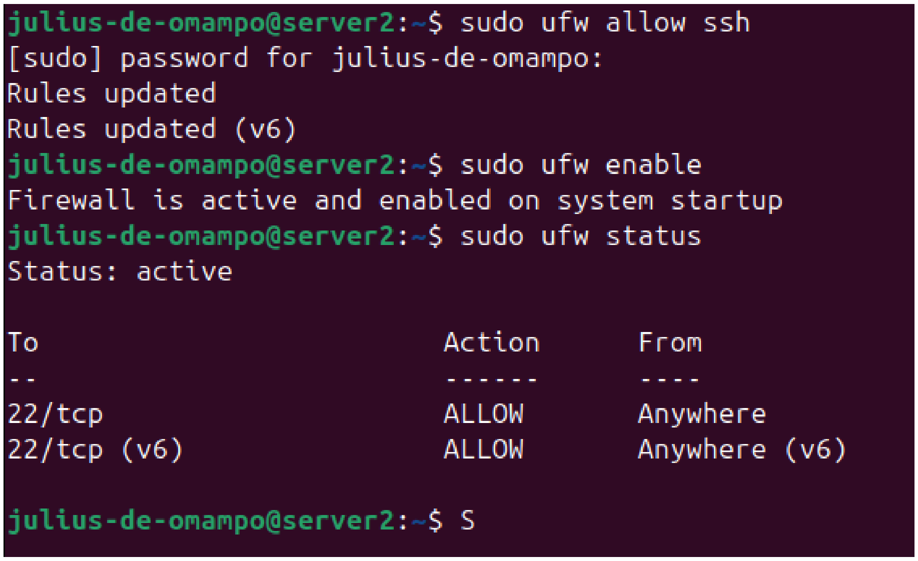
4. Configure the firewall to all port 22 by issuing the following commands:
    4.1 *sudo ufw allow ssh*
    4.2 *sudo ufw enable*
    4.3 *sudo ufw status*



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

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

julius-de-omampo@workstation:~$ 
```



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

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

julius-de-omampo@server1:~$
```

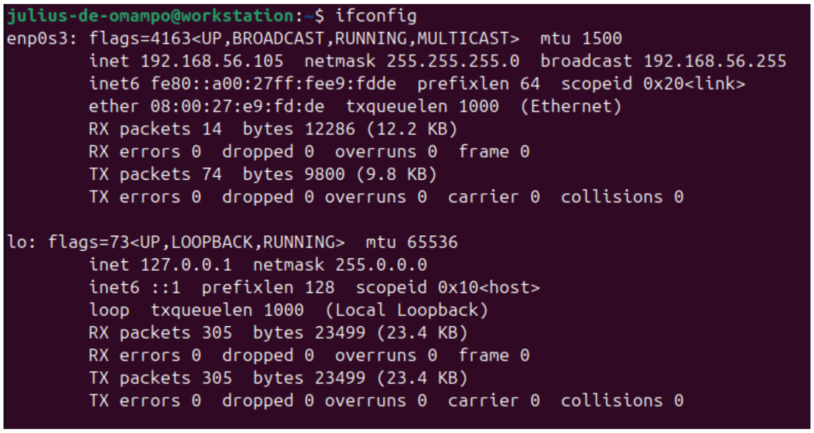


```
julius-de-omampo@server2:~$ sudo ufw allow ssh
[sudo] password for julius-de-omampo:
Rules updated
Rules updated (v6)
julius-de-omampo@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
julius-de-omampo@server2:~$ sudo ufw status
Status: active

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

julius-de-omampo@server2:~$ S
```

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

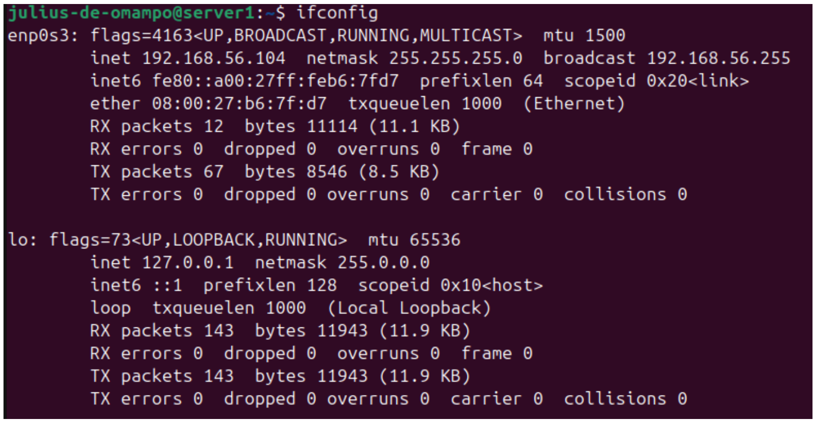
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.

   1.1 Local machine IP address: 192.168.56.**105**



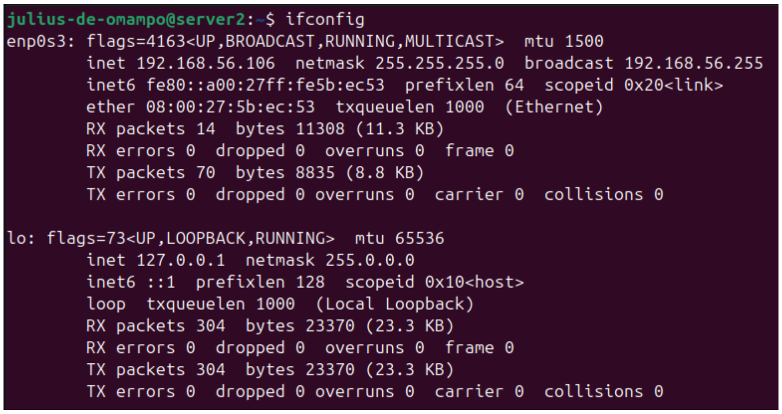
```
julius-de-omampo@workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.105  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fee9:fdde  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e9:fd:de  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 12286 (12.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 9800 (9.8 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 305  bytes 23499 (23.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 305  bytes 23499 (23.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

   1.2 Server 1 IP address: 192.168.56.**104**



```
julius-de-omampo@server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.104  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:feb6:7fd7  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:b6:7f:d7  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 11114 (11.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 67  bytes 8546 (8.5 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 143  bytes 11943 (11.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 143  bytes 11943 (11.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

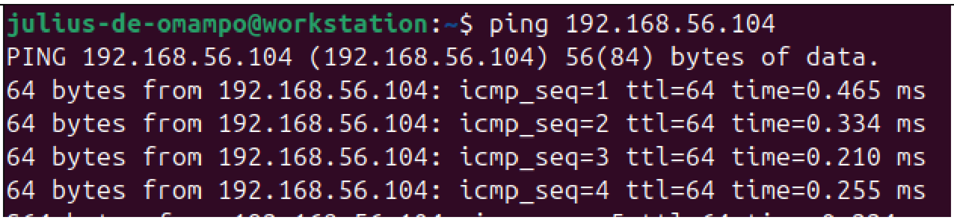
   1.3 Server 2 IP address: 192.168.56.**106**



```
julius-de-omampo@server2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.106  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe5b:ec53  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:5b:ec:53  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 11308 (11.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 70  bytes 8835 (8.8 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 304  bytes 23370 (23.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 304  bytes 23370 (23.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

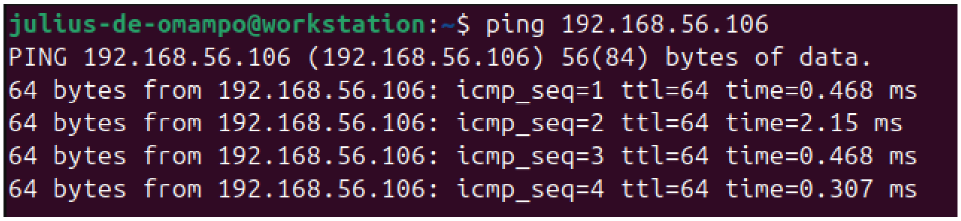
2. Make sure that they can ping each other.

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



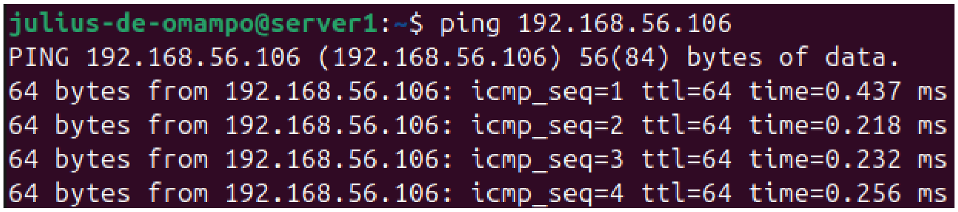
```
julius-de-omampo@workstation:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=0.465 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.334 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.210 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.255 ms
```

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



```
julius-de-omampo@workstation:~$ ping 192.168.56.106
PING 192.168.56.106 (192.168.56.106) 56(84) bytes of data.
64 bytes from 192.168.56.106: icmp_seq=1 ttl=64 time=0.468 ms
64 bytes from 192.168.56.106: icmp_seq=2 ttl=64 time=2.15 ms
64 bytes from 192.168.56.106: icmp_seq=3 ttl=64 time=0.468 ms
64 bytes from 192.168.56.106: icmp_seq=4 ttl=64 time=0.307 ms
```

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



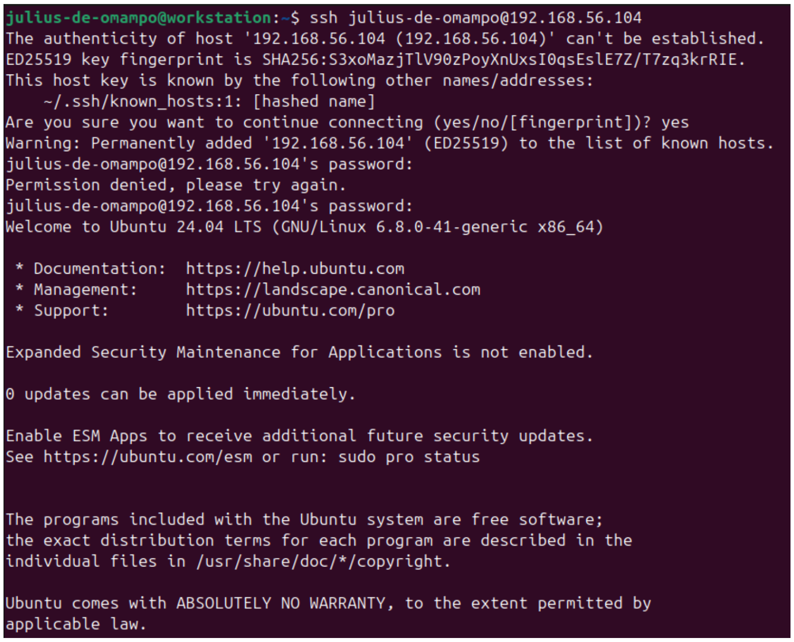
```
julius-de-omampo@server1:~$ ping 192.168.56.106
PING 192.168.56.106 (192.168.56.106) 56(84) bytes of data.
64 bytes from 192.168.56.106: icmp_seq=1 ttl=64 time=0.437 ms
64 bytes from 192.168.56.106: icmp_seq=2 ttl=64 time=0.218 ms
64 bytes from 192.168.56.106: icmp_seq=3 ttl=64 time=0.232 ms
64 bytes from 192.168.56.106: icmp_seq=4 ttl=64 time=0.256 ms
```

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

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

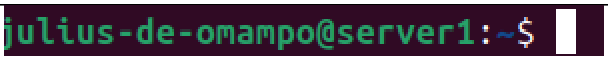
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted



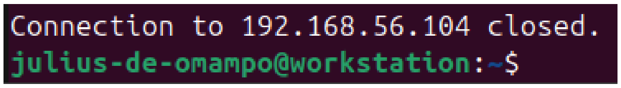
```
julius-de-omampo@workstation:~$ ssh julius-de-omampo@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:S3xoMazjTlV90zPoyXnUxsI0qsEslE7Z/T7zq3krRIE.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known hosts.
julius-de-omampo@192.168.56.104's password:
Permission denied, please try again.
julius-de-omampo@192.168.56.104's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-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.

0 updates can be applied immediately.

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


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.
```

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



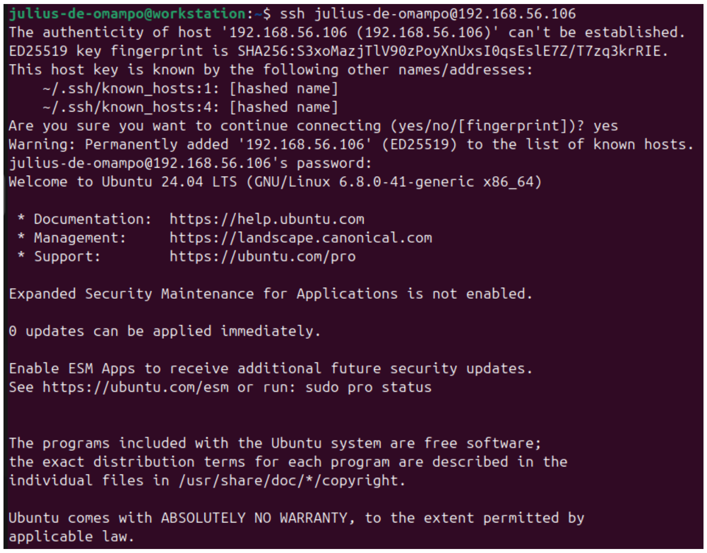
```
julius-de-omampo@server1:~$
```

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

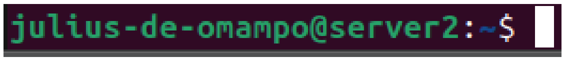


```
Connection to 192.168.56.104 closed.
julius-de-omampo@workstation:~$
```

3. Do the same for Server 2.



```
julius-de-omampo@workstation:~$ ssh julius-de-omampo@192.168.56.106
The authenticity of host '192.168.56.106 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:S3xoMazjTlV90zPoyXnUxsI0qsEslE7Z/T7zq3krRIE.
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 '192.168.56.106' (ED25519) to the list of known hosts.
julius-de-omampo@192.168.56.106's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-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.

0 updates can be applied immediately.

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


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.
```



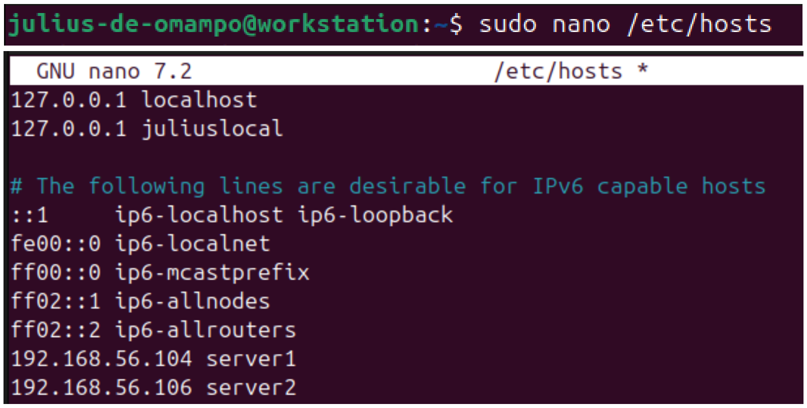
```
julius-de-omampo@server2:~$
```

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)

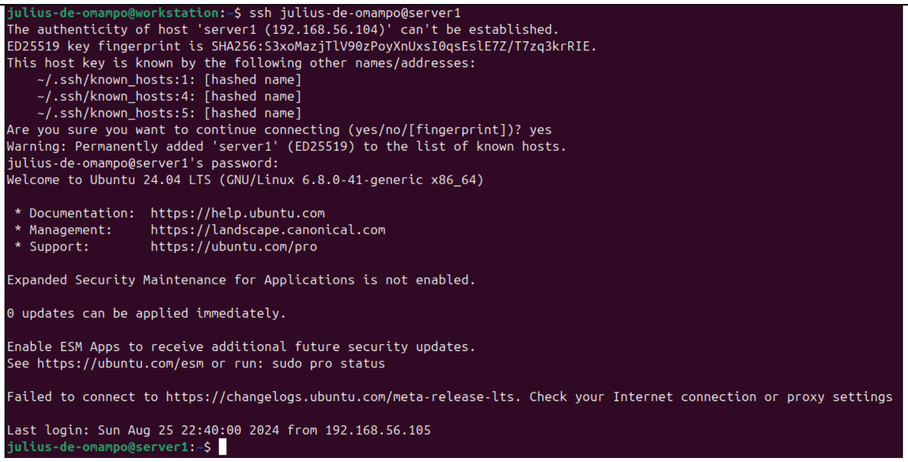
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

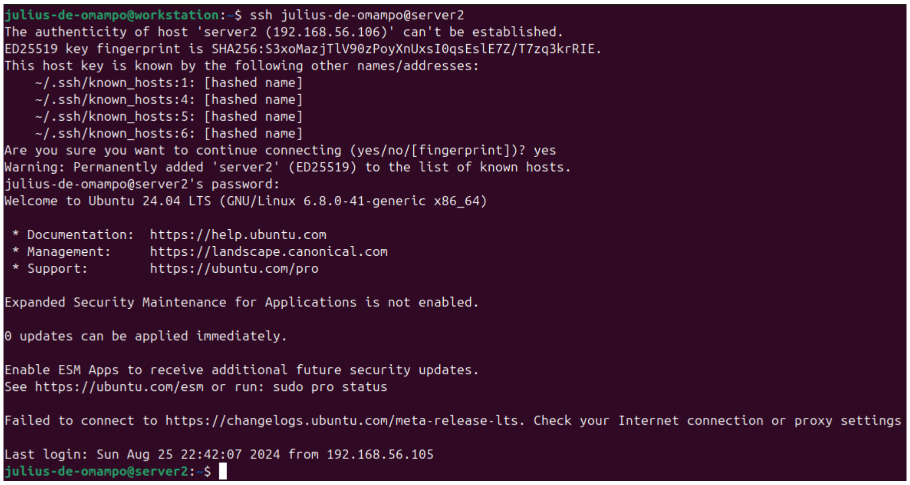


```
julius-de-omampo@workstation:~$ sudo nano /etc/hosts

  GNU nano 7.2                          /etc/hosts *
127.0.0.1 localhost
127.0.0.1 juliuslocal

# 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
192.168.56.104 server1
192.168.56.106 server2
```

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.



```
julius-de-omampo@workstation:~$ ssh julius-de-omampo@server1
The authenticity of host 'server1 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:S3xoMazjTlV90zPoyXnUxsI0qsEslE7Z/T7zq3krRIE.
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 'server1' (ED25519) to the list of known hosts.
julius-de-omampo@server1's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-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.

0 updates can be applied immediately.

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 22:40:00 2024 from 192.168.56.105
julius-de-omampo@server1:~$ 
```



```
julius-de-omampo@workstation:~$ ssh julius-de-omampo@server2
The authenticity of host 'server2 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:S3xoMazjTlV90zPoyXnUxsI0qsEslE7Z/T7zq3krRIE.
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]
    ~/.ssh/known_hosts:6: [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.
julius-de-omampo@server2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-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.

0 updates can be applied immediately.

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 22:42:07 2024 from 192.168.56.105
julius-de-omampo@server2:~$ 
```

Reflections:

Answer the following:

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

We were able to use the hostname instead of the IP address of the computer in SSH commands if the said hostname is resolvable. It can be done through the use of DNS (Domain Name System) resolution if the hostname is registered with a proper DNS server. It is done by mapping the hostname to an IP address in the '/etc/hosts' file for a local resolution. The aforementioned method allows the system to translate the specific hostname to the correct respective IP address, allowing us to use the computer's hostname directly in SSH commands.

2. How secured is SSH?

SSH (Secure Shell) is considered to possess a high security for remote access and management of systems. It encrypts all the data transmitted between a client and server, which includes: passwords, commands, and files, which protects against eavesdropping and data leaks. Moreover, SSH uses strong cryptographic algorithm such as AES for encryption and it uses various algorithms for key exchange and authentication. Additionally, SSH also supports multi-factor authentication, which adds additional layer of security. On the other hand, its security also depends on proper configuration, such as using strong passwords or key-based authentication, disabling root login, and regularly updating the software to protect against security breaches.