Name: John Edward Miles D. Espiritu	**Date Performed:01/16/2024**
Course/Section:CPE232-31S1	**Date Submitted:01/16/2024**
Instructor: Sir Jonathan Taylar	**Semester and SY: 2nd sem/2024-2025**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

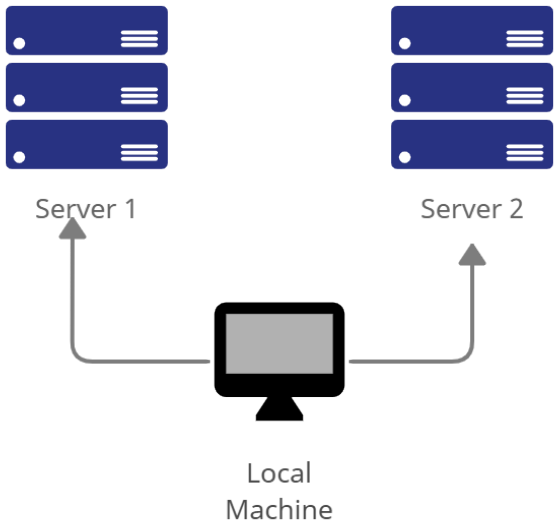
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

1.2 Use server2 for Server 2

```
  GNU nano 7.2                                /etc/hostname *
ControlNode2
```

1.3 Use workstation for the Local Machine

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

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 7.2                                /etc/hosts
127.0.0.1 ControlNode1
127.0.1.1 jem-espiritu-VirtualBox

# 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

```
  GNU nano 7.2                                /etc/hosts
127.0.0.1 ControlNode2
127.0.1.1 jem-espiritu-VirtualBox

# 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

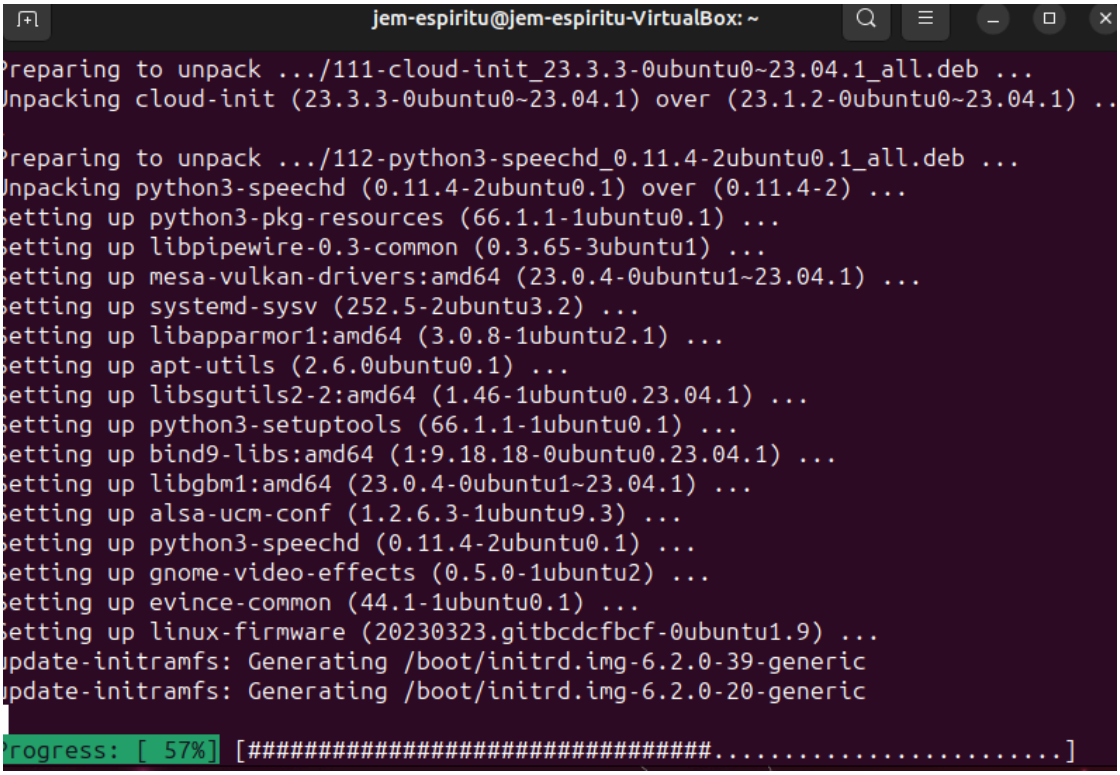
```
  GNU nano 7.2                                /etc/hosts *
127.0.0.1 ManagedNode
127.0.1.1 jem-espiritu-VirtualBox

# 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
```

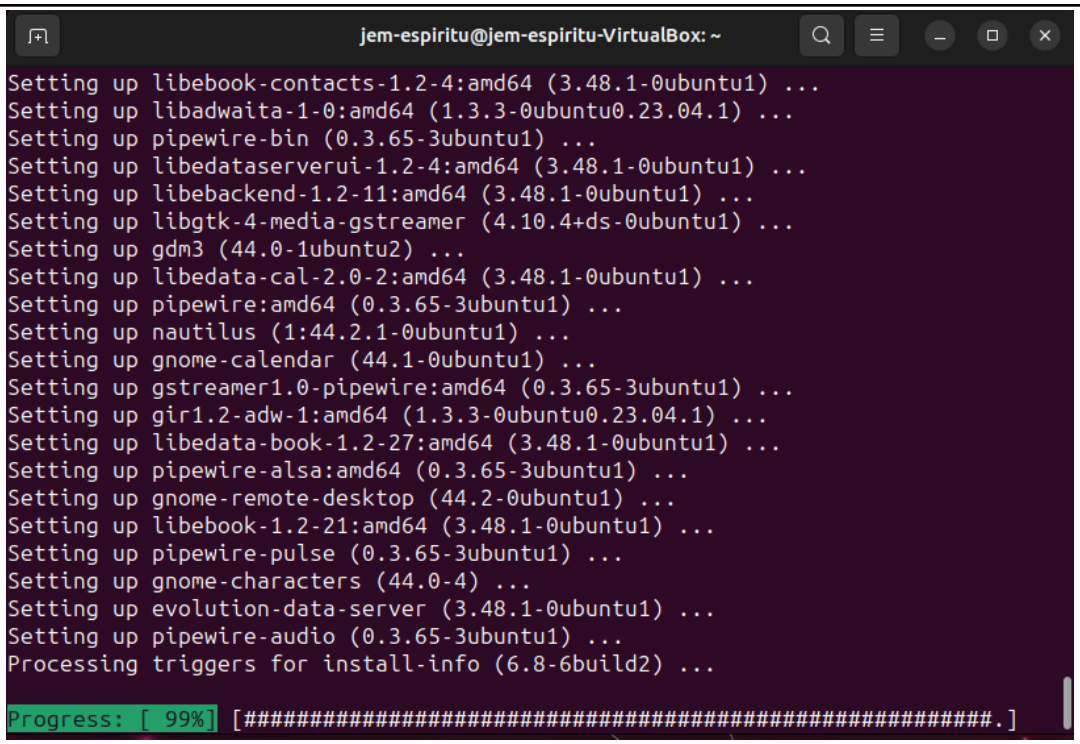
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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

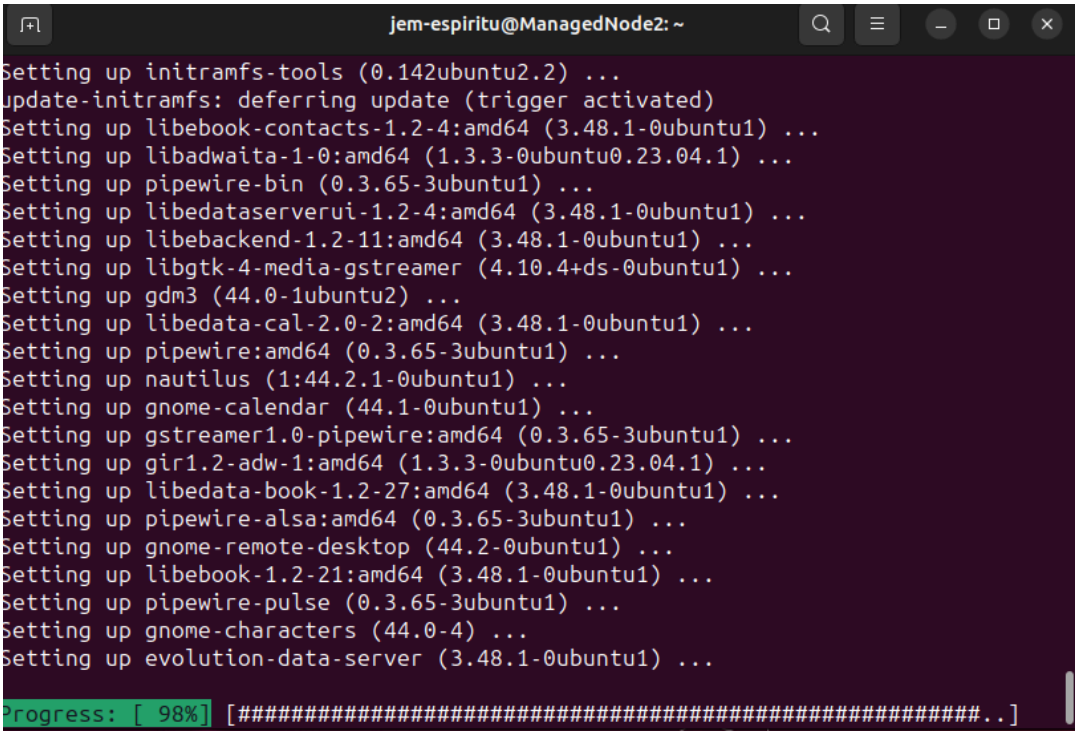
Local Machine



SERVER1

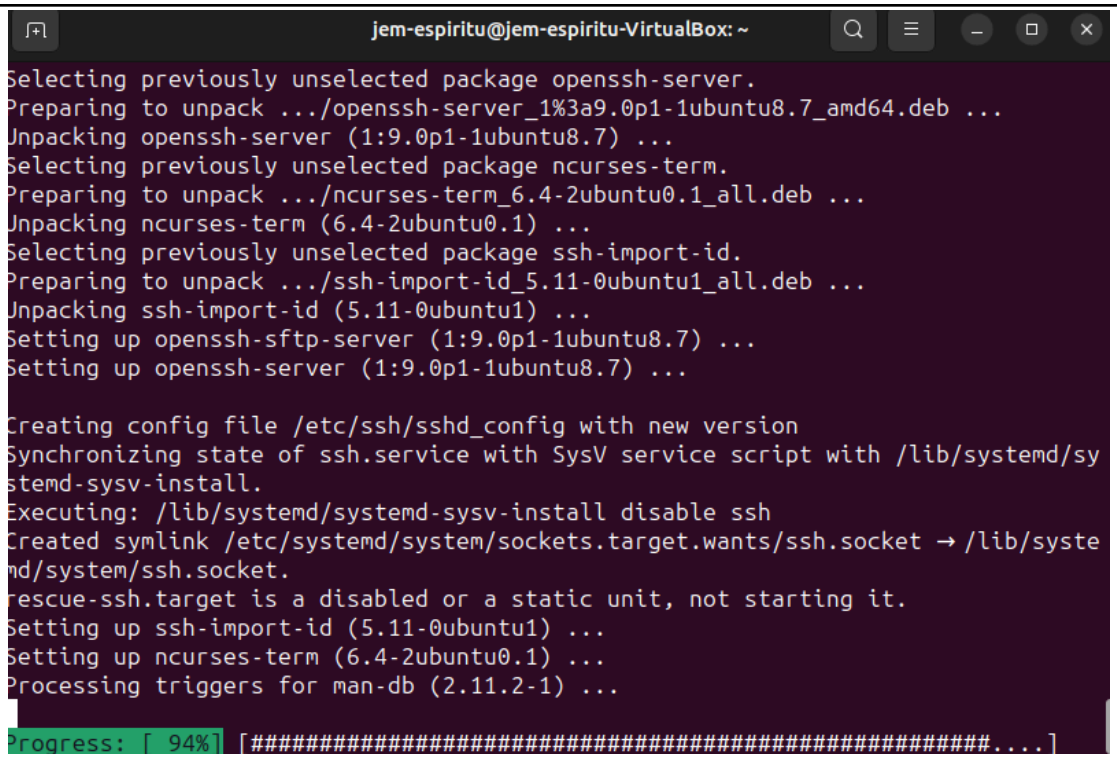


SERVER2

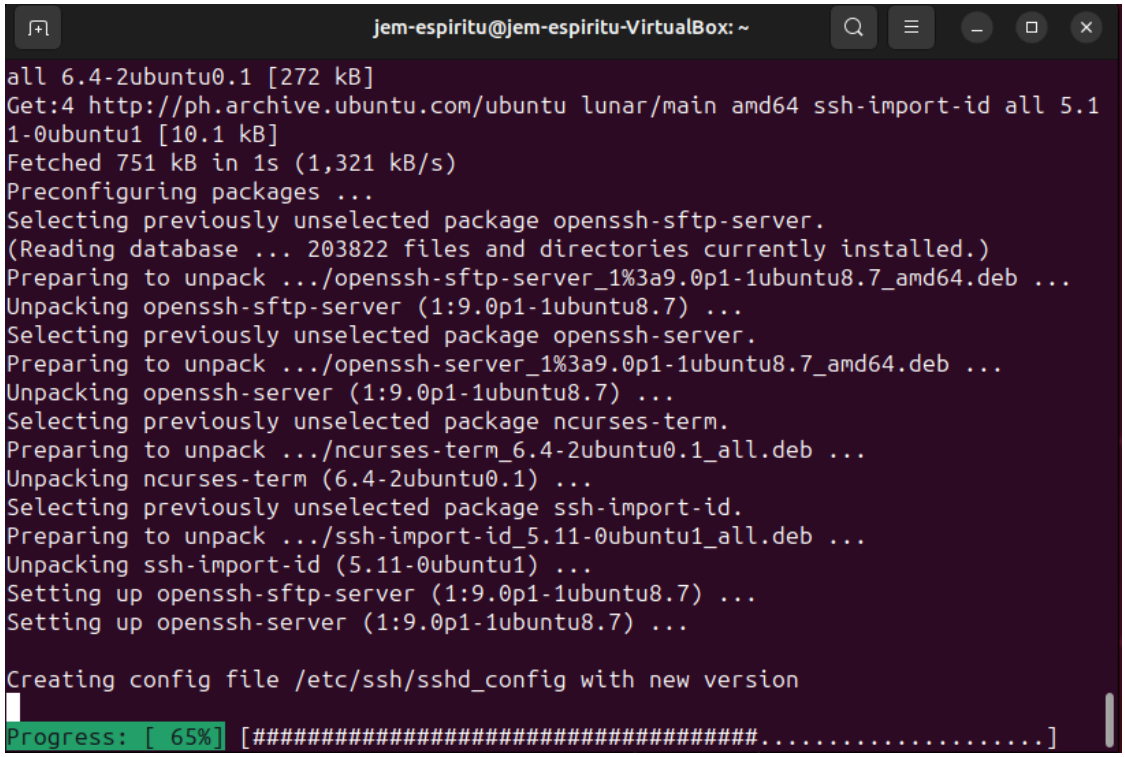


2. Install the SSH server using the command *sudo apt install openssh-server*.

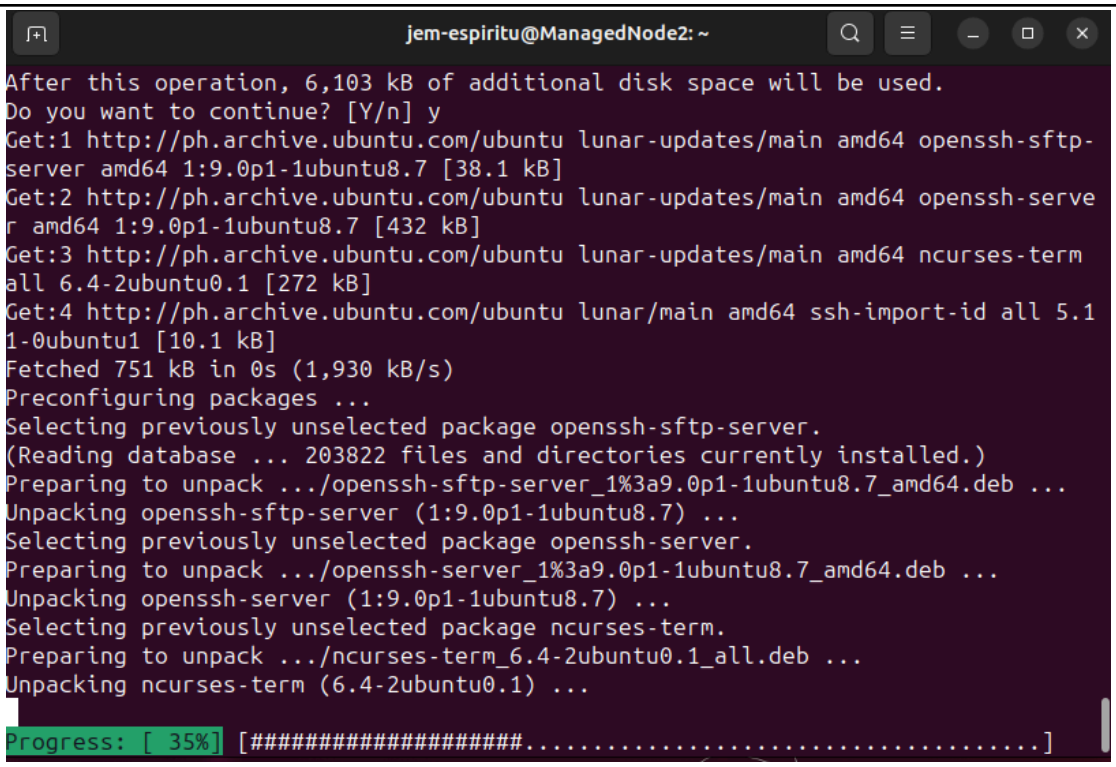
LOCALMACHINE



SERVER1

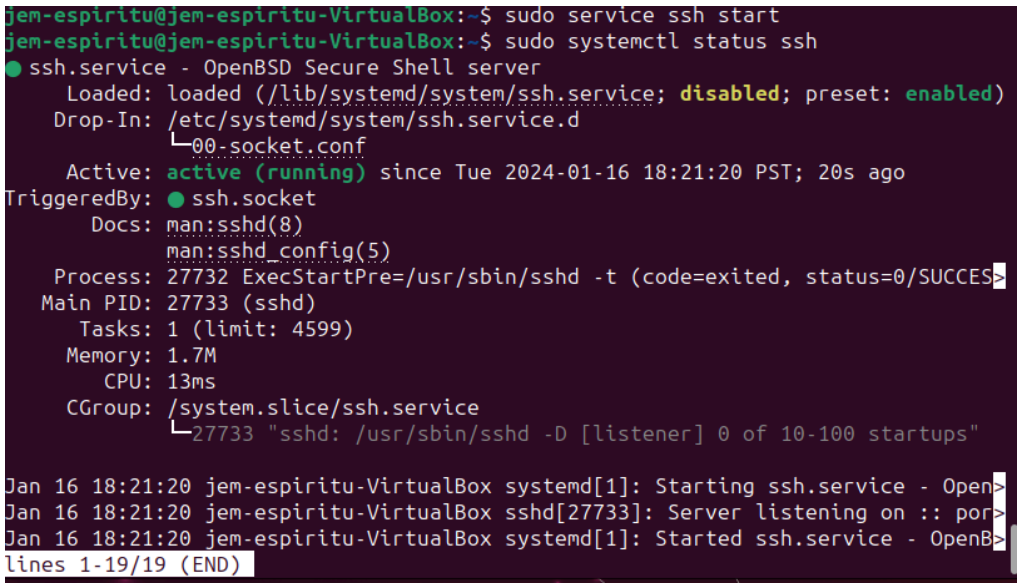


SERVER2



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*

LOCALMACHINE



SERVER1

```
jem-espiritu@jem-espiritu-VirtualBox:~$ sudo service ssh start
jem-espiritu@jem-espiritu-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
    Drop-In: /etc/systemd/system/ssh.service.d
             └─00-socket.conf
     Active: active (running) since Tue 2024-01-16 18:23:03 PST; 8s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 27756 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 27757 (sshd)
      Tasks: 1 (limit: 4599)
     Memory: 1.8M
        CPU: 13ms
     CGroup: /system.slice/ssh.service
             └─27757 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

SERVER2

```
jem-espiritu@ManagedNode2:~$ sudo service ssh start
jem-espiritu@ManagedNode2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
    Drop-In: /etc/systemd/system/ssh.service.d
             └─00-socket.conf
     Active: active (running) since Tue 2024-01-16 18:24:39 PST; 7s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 26881 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 26882 (sshd)
      Tasks: 1 (limit: 4599)
     Memory: 1.7M
        CPU: 13ms
     CGroup: /system.slice/ssh.service
             └─26882 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 16 18:24:39 ManagedNode2 sshd[26882]: Server listening on :: port 22.
Jan 16 18:24:39 ManagedNode2 systemd[1]: Started ssh.service - OpenBSD Secure S>
```

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*

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: 192.168.56.105
   1.2 Server 2 IP address: 192.168.56.106
   1.3 Server 3 IP address: 192.168.56.107
2. Make sure that they can ping each other.

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

```
jem-espiritu@ManagedNode:~$ 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=11.5 ms
64 bytes from 192.168.56.106: icmp_seq=2 ttl=64 time=3.30 ms
64 bytes from 192.168.56.106: icmp_seq=3 ttl=64 time=0.445 ms
64 bytes from 192.168.56.106: icmp_seq=4 ttl=64 time=1.71 ms
64 bytes from 192.168.56.106: icmp_seq=5 ttl=64 time=2.57 ms
64 bytes from 192.168.56.106: icmp_seq=6 ttl=64 time=3.37 ms
64 bytes from 192.168.56.106: icmp_seq=7 ttl=64 time=2.10 ms
64 bytes from 192.168.56.106: icmp_seq=8 ttl=64 time=0.365 ms
```

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

```
jem-espiritu@ManagedNode:~$ ping 192.168.56.107
PING 192.168.56.107 (192.168.56.107) 56(84) bytes of data.
64 bytes from 192.168.56.107: icmp_seq=1 ttl=64 time=6.74 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.891 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=24.9 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=2.47 ms
64 bytes from 192.168.56.107: icmp_seq=5 ttl=64 time=0.674 ms
64 bytes from 192.168.56.107: icmp_seq=6 ttl=64 time=0.704 ms
64 bytes from 192.168.56.107: icmp_seq=7 ttl=64 time=3.36 ms
64 bytes from 192.168.56.107: icmp_seq=8 ttl=64 time=1.14 ms
64 bytes from 192.168.56.107: icmp_seq=9 ttl=64 time=0.591 ms
64 bytes from 192.168.56.107: icmp_seq=10 ttl=64 time=0.772 ms
64 bytes from 192.168.56.107: icmp_seq=11 ttl=64 time=0.784 ms
```

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

```
jem-espiritu@ControlNode1:~$ ping 192.168.56.107
PING 192.168.56.107 (192.168.56.107) 56(84) bytes of data.
64 bytes from 192.168.56.107: icmp_seq=1 ttl=64 time=0.719 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.611 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=0.880 ms
64 bytes from 192.168.56.107: icmp_seq=5 ttl=64 time=0.425 ms
64 bytes from 192.168.56.107: icmp_seq=6 ttl=64 time=0.328 ms
64 bytes from 192.168.56.107: icmp_seq=7 ttl=64 time=0.323 ms
64 bytes from 192.168.56.107: icmp_seq=8 ttl=64 time=0.305 ms
64 bytes from 192.168.56.107: icmp_seq=9 ttl=64 time=0.375 ms
64 bytes from 192.168.56.107: icmp_seq=10 ttl=64 time=0.391 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

```
jem-espiritu@ManagedNode:~$ ssh jem-espiritu@192.168.56.106
jem-espiritu@192.168.56.106's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

0 updates can be applied immediately.

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 16 18:51:11 2024 from 192.168.56.106
jem-espiritu@ControlNode1:~$
```

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

```
jem-espiritu@ManagedNode:~$ ssh jem-espiritu@192.168.56.106
jem-espiritu@192.168.56.106's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

0 updates can be applied immediately.

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 16 18:51:11 2024 from 192.168.56.106
jem-espiritu@ControlNode1:~$
```

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

```
jem-espiritu@ControlNode1:~$
logout
Connection to 192.168.56.106 closed.
jem-espiritu@ManagedNode:~$
```

3. Do the same for Server 2.

```
jem-espiritu@ControlNode2:~$ ssh jem-espiritu@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:j/xqqGtA2aq5ouzdtpKwd3+psnHYcUuqXigDNLA1E1s.
This key is not known by any other names
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
.
jem-espiritu@192.168.56.106's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

0 updates can be applied immediately.

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 16 18:55:51 2024 from 192.168.56.105
jem-espiritu@ControlNode1:~$
```

```
jem-espiritu@ControlNode1:~$
logout
Connection to 192.168.56.106 closed.
jem-espiritu@ControlNode2:~$
```

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

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

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

4.3 Save the file and exit.

```
jem-espiritu@ManagedNode: ~
  GNU nano 7.2                          /etc/hosts *
192.168.56.106 ControlNode1
192.168.56.107 ControlNode2

# 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.

```
jem-espiritu@ManagedNode:~$ ssh jem-espiritu@ControlNode1
The authenticity of host 'controlnode1 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:j/xqqGtA2aq5ouzdtpKwd3+psnHYcUuqXigDNLA1E1s.
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 'controlnode1' (ED25519) to the list of known hosts.
jem-espiritu@controlnode1's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

0 updates can be applied immediately.

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 16 18:58:51 2024 from 192.168.56.107
jem-espiritu@ControlNode1:~$
```

```
jem-espiritu@ManagedNode:~$ ssh jem-espiritu@ControlNode2
The authenticity of host 'controlnode2 (192.168.56.107)' can't be established.
ED25519 key fingerprint is SHA256:j/xqqGtA2aq5ouzdtpKwd3+psnHYcUuqXigDNLA1E1s.
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 'controlnode2' (ED25519) to the list of known hosts.
jem-espiritu@controlnode2's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

0 updates can be applied immediately.


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.

jem-espiritu@ControlNode2:~$
```

Reflections:

Answer the following:

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

To use a hostname instead of an IP address in SSH commands, you need to add an entry for the hostname and its corresponding IP address in the /etc/hosts file 12.

Alternatively, SSH configuration file ~/.ssh/config This will allow you to use the hostname instead of the IP address when connecting to the server via SSH.

2. How secured is SSH?

SSH (Secure Shell) is a protocol that provides secure communication between two networked devices It uses encryption to protect the data being transferred and provides password or public-key based authentication SSH is considered to be a secure alternative to legacy login protocols such as Telnet and Rlogin, and insecure file transfer methods such as FTP .The security of an SSH connection is only as good as its endpoints, meaning that if someone or something has low-level access to the operating system or network, it is possible to hook into the carrier and read the content of the tunnel. However, SSH is key-based authentication that is not prone to brute-force attacks, making it more secure than login IDs and passwords . There is no exposure of valid credentials if a server has been compromised. In summary, SSH is a secure protocol that provides secure communication between two networked devices, but its security is only as good as its endpoints.