Name: Andreu John L. Salvador	Date Performed: 22/08/2023
Course/Section:CPE 232–CPE31S5	Date Submitted: 23/08/2023
Instructor: Engr. Roman Richard	Semester and SY: 2nd 2023-2024

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

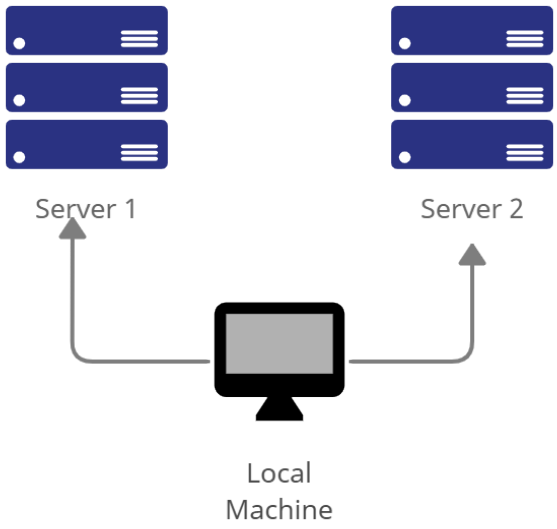
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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*).

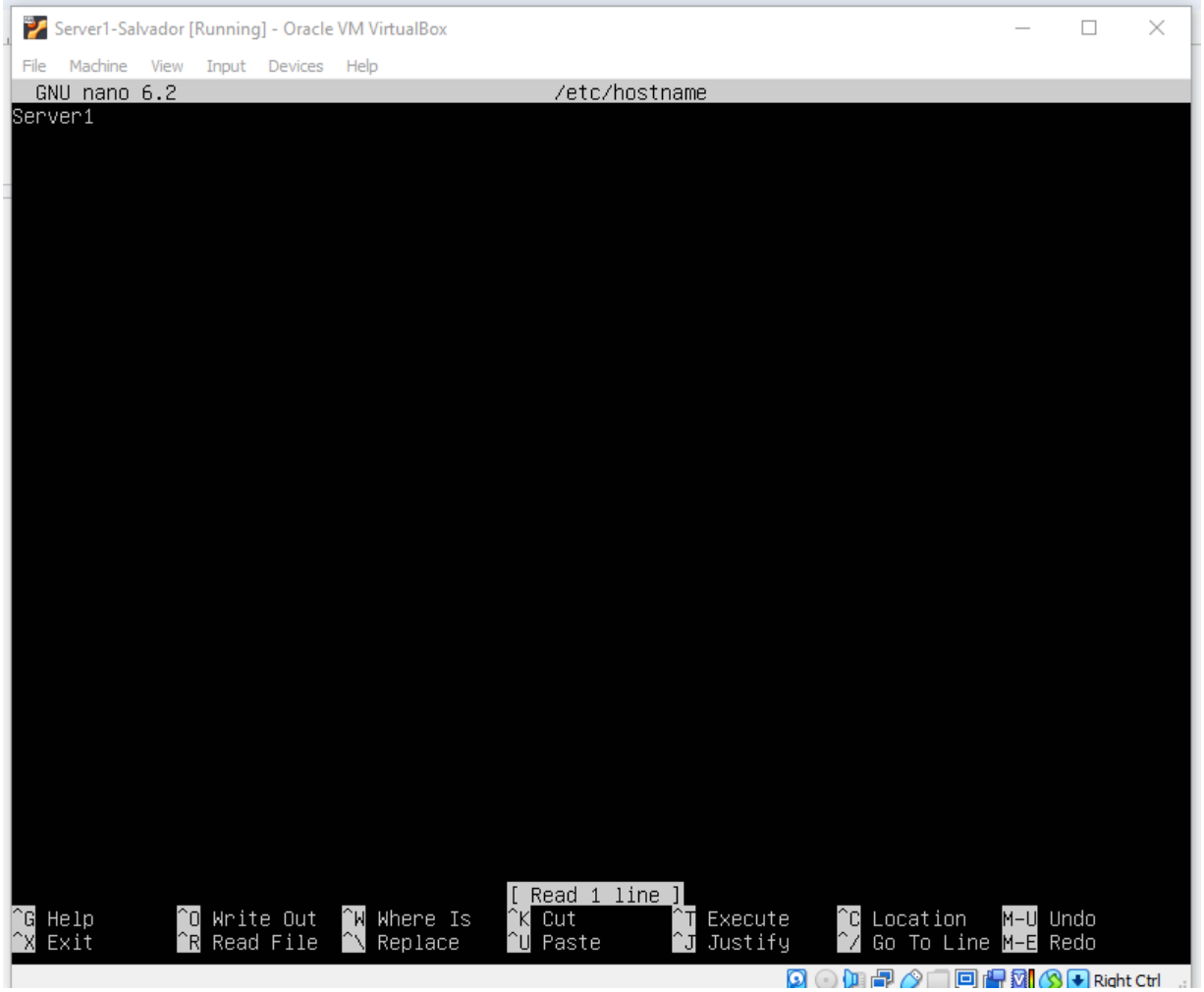


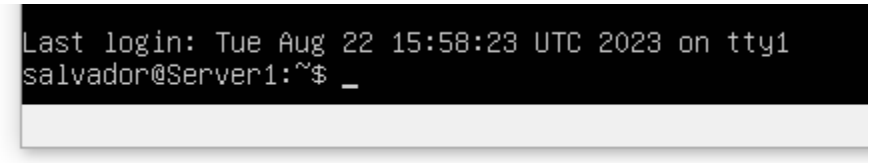
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

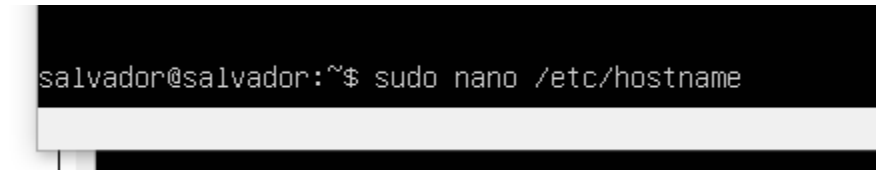
1.1 Use server1 for Server 1

```
salvador@salvador:~$ sudo nano /etc/hostname_
```

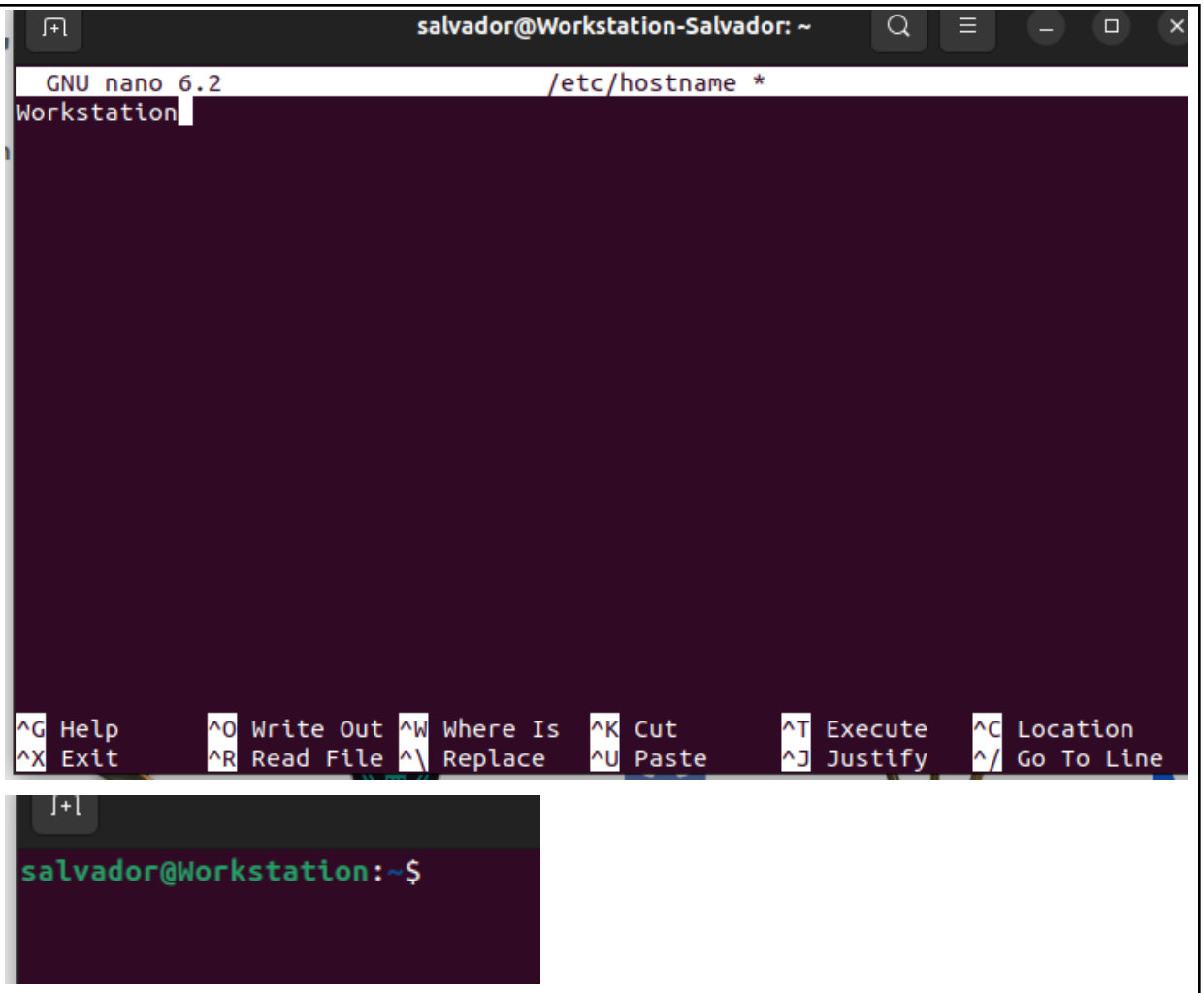




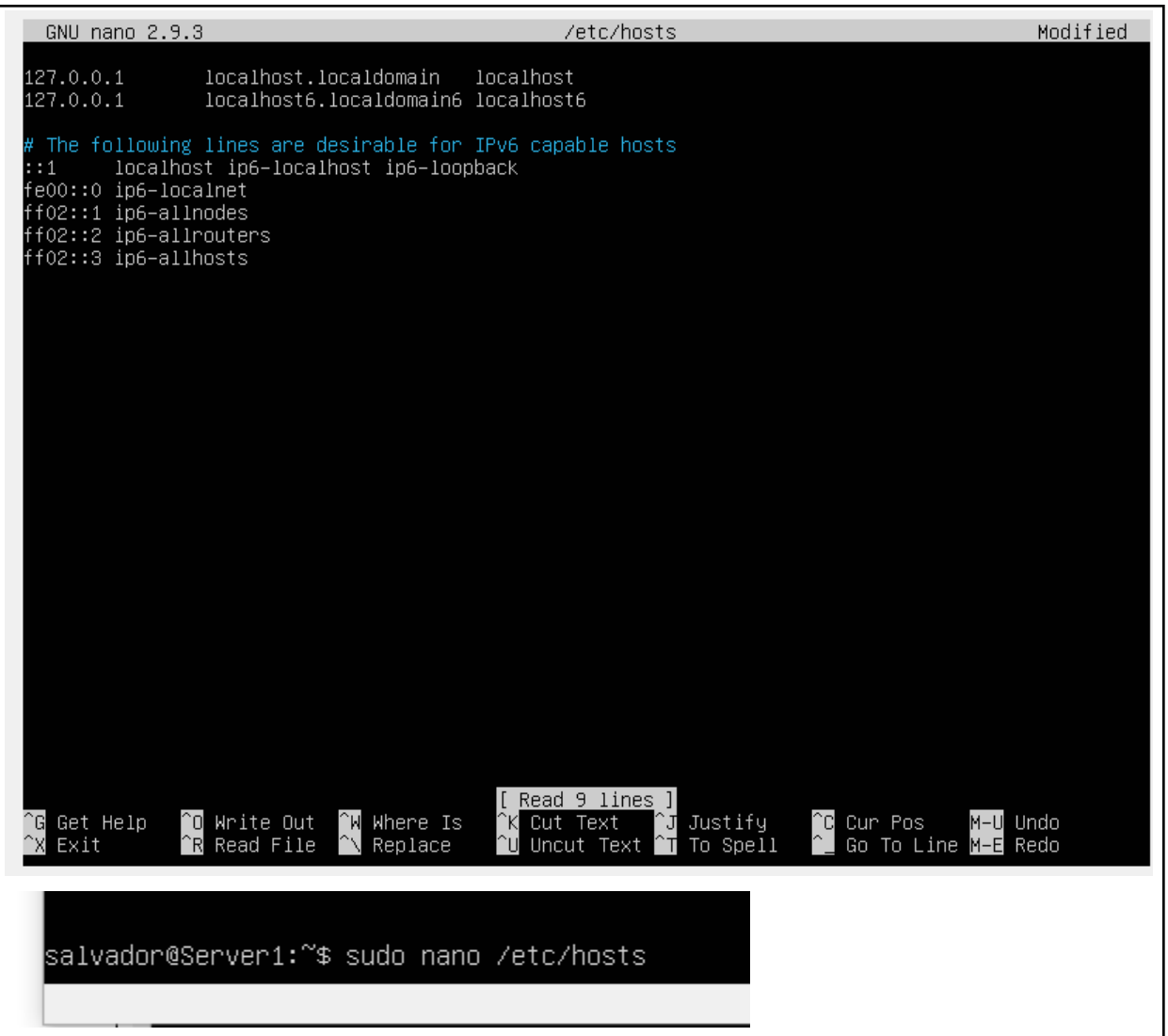
1.2 Use server2 for Server 2



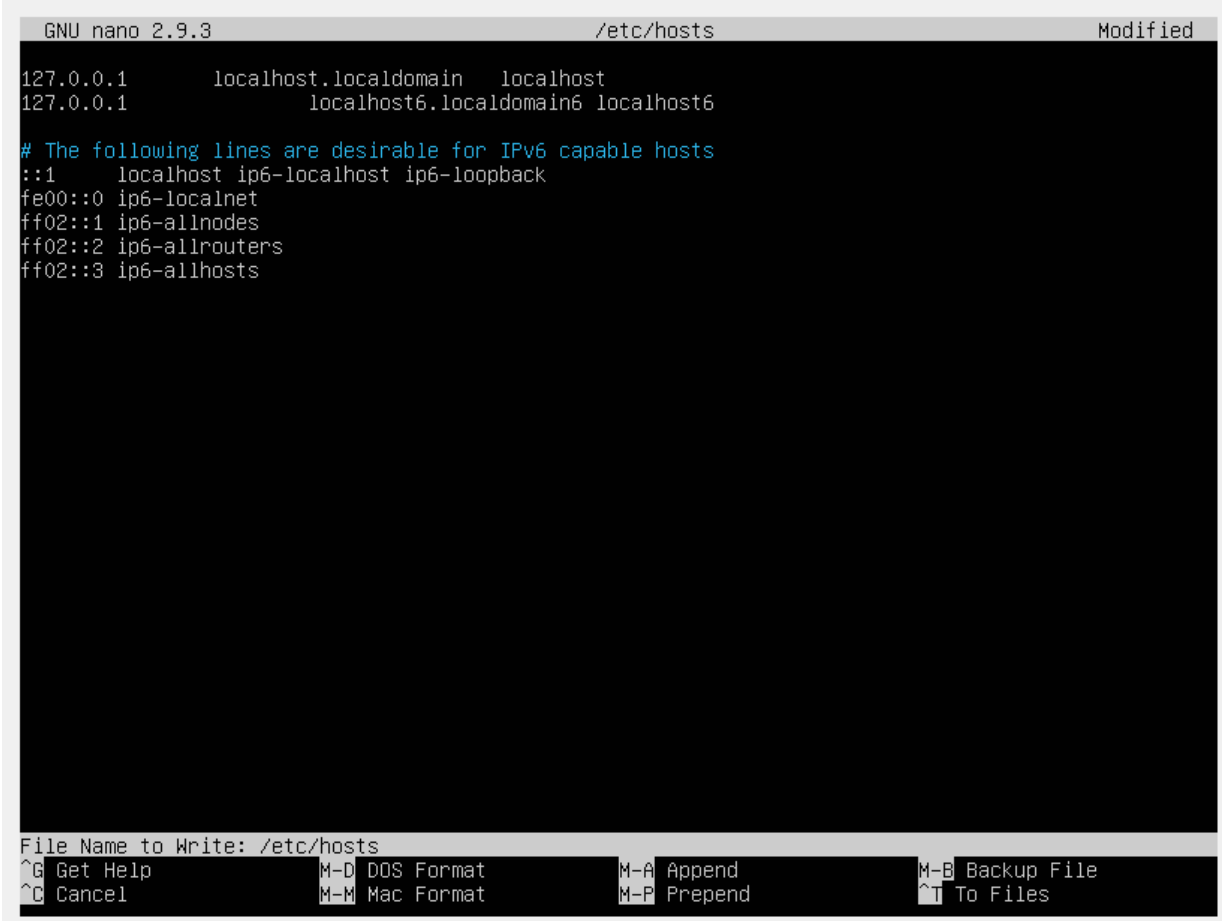
1.3 Use workstation for the Local Machine

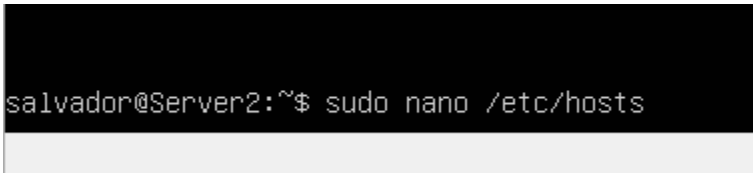


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1

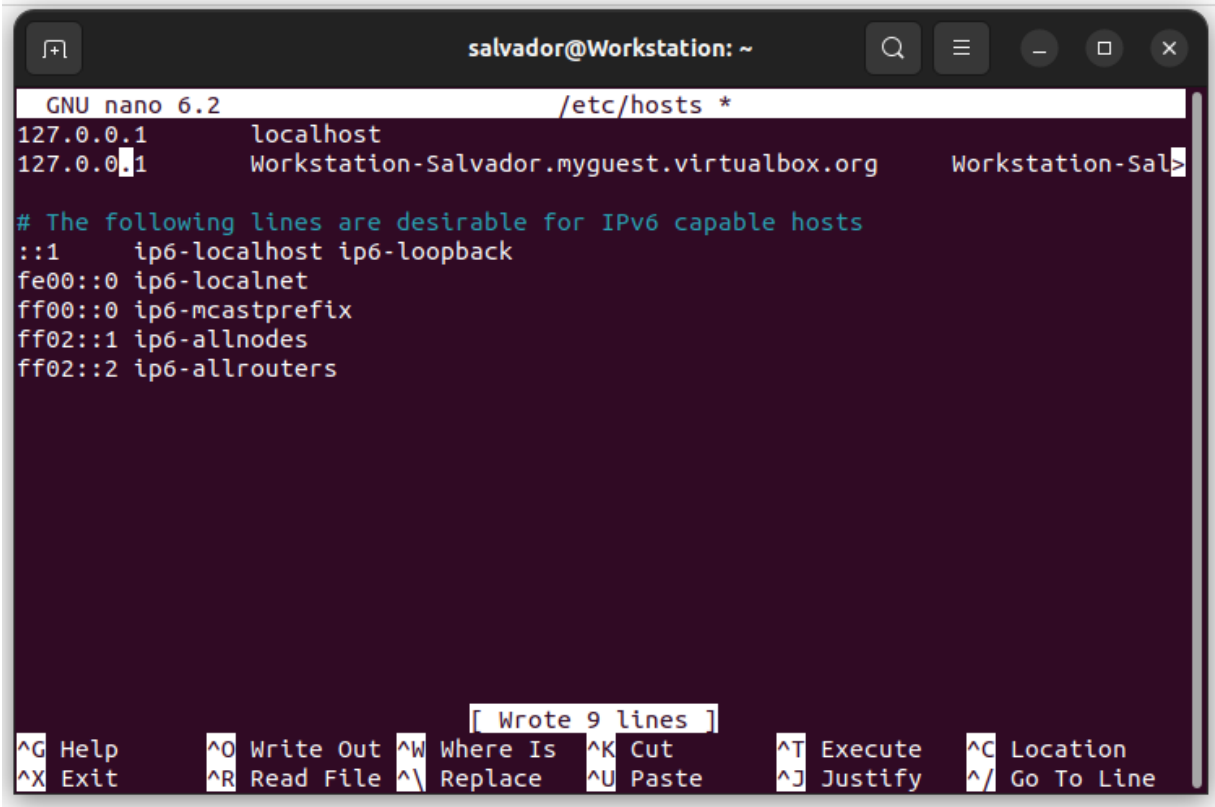


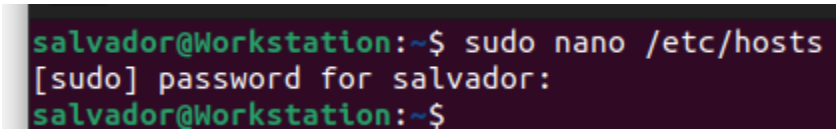
2.2 Type 127.0.0.1 server 2 for Server 2





2.3 Type 127.0.0.1 workstation for the Local Machine





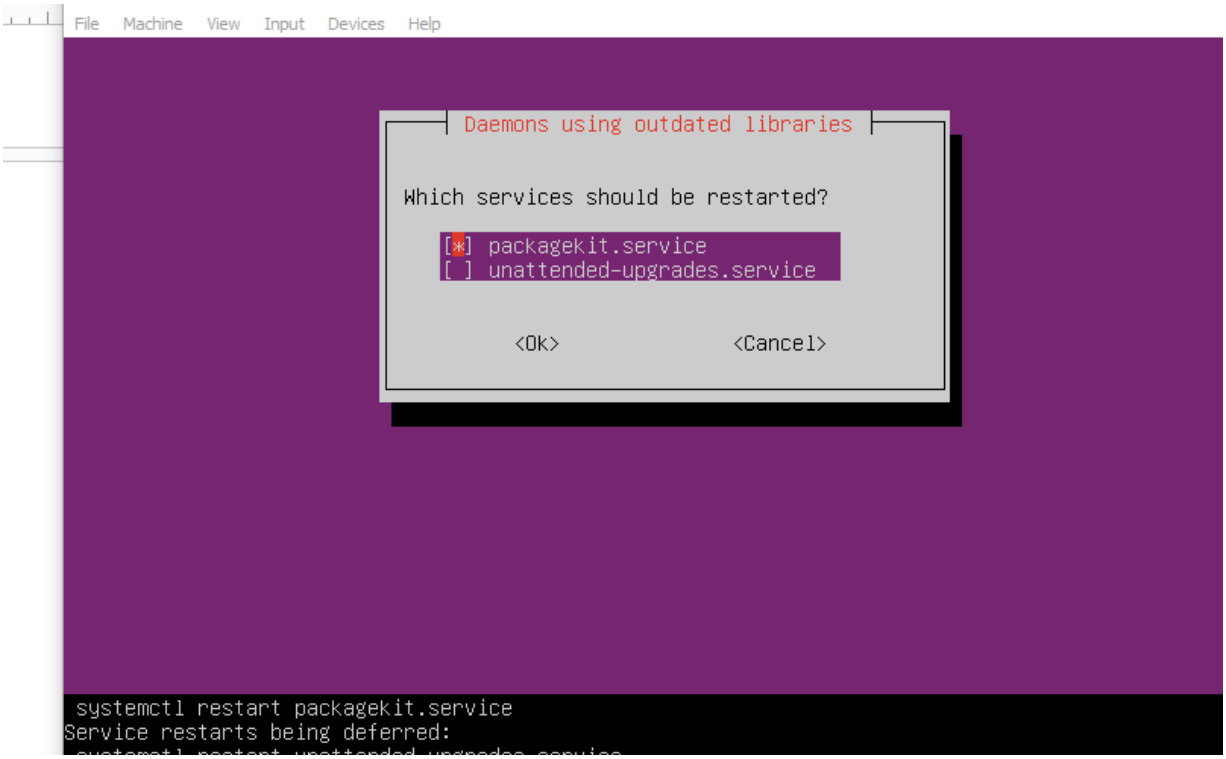
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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

Server 1:

```
salvador@Server1:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [894 kB]
Fetched 1,232 kB in 2s (646 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
salvador@Server1:~$ _
```

```
13 packages can be upgraded. Run 'apt list -
salvador@Server1:~$ sudo apt upgrade_
```



Server 2:

```
salvador@Server2:~$ sudo apt update
```

```
salvador@Server2:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [894 kB]
Fetched 1,232 kB in 2s (658 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
salvador@Server2:~$ _
```

```
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgr
salvador@Server2:~$ sudo apt upgrade_
```

File Machine View Input Devices Help

Daemons using outdated libraries

Which services should be restarted?

[*] packagekit.service
[ ] unattended-upgrades.service

<Ok> <Cancel>

```
systemctl restart packagekit.service
Service restarts being deferred:
systemctl restart unattended-upgrades.service
```

Workstation:

```
salvador@Workstation:~$ sudo apt update
```

```
Reading package lists... 8%
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
salvador@Workstation:~$
```

```
salvador@Workstation:~$ sudo apt upgrade
```

```
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for initramfs-tools (0.140ubuntu13.2) ...
update-initramfs: Generating /boot/initrd.img-6.2.0-26-generic
salvador@Workstation:~$
```

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

Server1:

```
No VM guests are running outdated hypervisor (qemu) binaries on
salvador@Server1:~$ sudo apt install openssh-server_
```

```
salvador@Server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
salvador@Server1:~$
```

Server2:

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
salvador@Server2:~$ sudo apt install openssh-server_
```

```
salvador@Server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
salvador@Server2:~$
```

Workstation:

```
update-initramfs: Generating /boot/initrd.img-6.2.0-26-gene
salvador@Workstation:~$ sudo apt install openssh-server
Reading package lists... Done
```

```
ce.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh
ystemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
salvador@Workstation:~$
```

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*

Server1:

```
0 upgraded, 0 newly installed, 0 to remove a
salvador@Server1:~$ sudo service ssh start
[sudo] password for salvador:
salvador@Server1:~$
```

```
salvador@Server1:~$ sudo service ssh start
[sudo] password for salvador:
salvador@Server1:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-22 22:03:25 UTC; 49min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 691 (sshd)
      Tasks: 1 (limit: 2964)
     Memory: 4.3M
        CPU: 50ms
     CGroup: /system.slice/ssh.service
             └─691 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 22:03:21 Server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 22:03:25 Server1 sshd[691]: Server listening on 0.0.0.0 port 22.
Aug 22 22:03:25 Server1 sshd[691]: Server listening on :: port 22.
Aug 22 22:03:25 Server1 systemd[1]: Started OpenBSD Secure Shell server.
salvador@Server1:~$ _
```

Server2:

```
salvador@Server2:~$ sudo service ssh start
salvador@Server2:~$
```

```
salvador@Server2:~$ sudo service ssh start
salvador@Server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-22 22:42:18 UTC; 11min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 692 (sshd)
      Tasks: 1 (limit: 2964)
     Memory: 4.4M
        CPU: 71ms
     CGroup: /system.slice/ssh.service
             └─692 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 22:42:15 Server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 22:42:18 Server2 sshd[692]: Server listening on 0.0.0.0 port 22.
Aug 22 22:42:18 Server2 sshd[692]: Server listening on :: port 22.
Aug 22 22:42:18 Server2 systemd[1]: Started OpenBSD Secure Shell server.
salvador@Server2:~$
```

Workstation:

```
lines 1-16/16 (END)
salvador@Workstation:~$ sudo service ssh start
salvador@Workstation:~$
```

```
salvador@Workstation:~$ 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 Wed 2023-08-23 07:01:52 +08; 1min 45s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 9197 (sshd)
      Tasks: 1 (limit: 3460)
     Memory: 1.7M
        CPU: 33ms
     CGroup: /system.slice/ssh.service
             └─9197 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 23 07:01:52 Workstation systemd[1]: Starting OpenBSD Secure Shell server...
Aug 23 07:01:52 Workstation sshd[9197]: Server listening on 0.0.0.0 port 22.
Aug 23 07:01:52 Workstation sshd[9197]: Server listening on :: port 22.
Aug 23 07:01:52 Workstation systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (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*

Server1:

```
salvador@Server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
salvador@Server1:~$
```

```
salvador@Server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
salvador@Server1:~$
```

```
salvador@Server1:~$ sudo ufw status
Status: active

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

Server2:

```
salvador@Server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
salvador@Server2:~$ _
```

```
Rules updated (v6)
salvador@Server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
salvador@Server2:~$
```

```
salvador@Server2:~$ sudo ufw status
Status: active

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

salvador@Server2:~$ _
```

Workstation:

```
lines 1-16/16 (END)
salvador@Workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

```
Rules updated (v6)
salvador@Workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

```
salvador@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 Server 1 IP address: 192.168.56.101

```
salvador@Server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe09:beba  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:09:be:ba  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 1180 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1334 (1.3 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 84  bytes 6220 (6.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 6220 (6.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

   1.2 Server 2 IP address: 192.168.56.102

```
salvador@Server2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fead:3c3f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ad:3c:3f  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 1180 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1334 (1.3 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 84  bytes 6220 (6.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 6220 (6.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

1.3 Server 3 IP address: 192.168.56.103

```
salvador@Workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.103  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::5ef7:4bb3:b159:1f85  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:10:cd:27  txqueuelen 1000  (Ethernet)
        RX packets 3  bytes 1770 (1.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 65  bytes 8859 (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 724  bytes 53684 (53.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 724  bytes 53684 (53.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

salvador@Workstation:~$
```

2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ✅ Successful ☐ Not Successful

```
salvador@Workstation:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.638 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.669 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.746 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.476 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.695 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.667 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.734 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.465 ms
64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.663 ms
64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.737 ms
64 bytes from 192.168.56.101: icmp_seq=11 ttl=64 time=0.709 ms
64 bytes from 192.168.56.101: icmp_seq=12 ttl=64 time=0.666 ms
64 bytes from 192.168.56.101: icmp_seq=13 ttl=64 time=0.653 ms
^C
--- 192.168.56.101 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12204ms
rtt min/avg/max/mdev = 0.465/0.655/0.746/0.085 ms
```

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

```
salvador@Workstation: ~
--- 192.168.56.101 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12204ms
rtt min/avg/max/mdev = 0.465/0.655/0.746/0.085 ms
salvador@Workstation:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=2.08 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.699 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.895 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.909 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.690 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.521 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.516 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.748 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.399 ms
64 bytes from 192.168.56.102: icmp_seq=10 ttl=64 time=0.701 ms
64 bytes from 192.168.56.102: icmp_seq=11 ttl=64 time=0.748 ms
64 bytes from 192.168.56.102: icmp_seq=12 ttl=64 time=0.461 ms
64 bytes from 192.168.56.102: icmp_seq=13 ttl=64 time=0.676 ms
64 bytes from 192.168.56.102: icmp_seq=14 ttl=64 time=0.784 ms
^C
--- 192.168.56.102 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13203ms
rtt min/avg/max/mdev = 0.399/0.773/2.078/0.390 ms
```

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

```
salvador@Server1:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.469 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.751 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.644 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.701 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.698 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.387 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=1.09 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.670 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.704 ms
64 bytes from 192.168.56.102: icmp_seq=10 ttl=64 time=0.702 ms
64 bytes from 192.168.56.102: icmp_seq=11 ttl=64 time=0.434 ms
64 bytes from 192.168.56.102: icmp_seq=12 ttl=64 time=0.624 ms
64 bytes from 192.168.56.102: icmp_seq=13 ttl=64 time=0.762 ms
64 bytes from 192.168.56.102: icmp_seq=14 ttl=64 time=0.686 ms
64 bytes from 192.168.56.102: icmp_seq=15 ttl=64 time=0.703 ms
^C
--- 192.168.56.102 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14077ms
rtt min/avg/max/mdev = 0.387/0.668/1.089/0.158 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*

```
salvador@Workstation:~$ ssh salvador@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:3Hfw5vuOnI0D+OpuV+naVNPpUjqNCNVEqZx6Wh08qSQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts
.
```

1.2 Enter the password for server 1 when prompted

```
salvador@192.168.56.101's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

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

```
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 11:42:01 PM UTC 2023

  System load:  0.046875          Processes:               104
  Usage of /:   42.8% of 12.31GB  Users logged in:         1
  Memory usage: 8%                IPv4 address for enp0s3: 192.168.56.101
  Swap usage:   0%


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


Last login: Tue Aug 22 23:19:02 2023
```

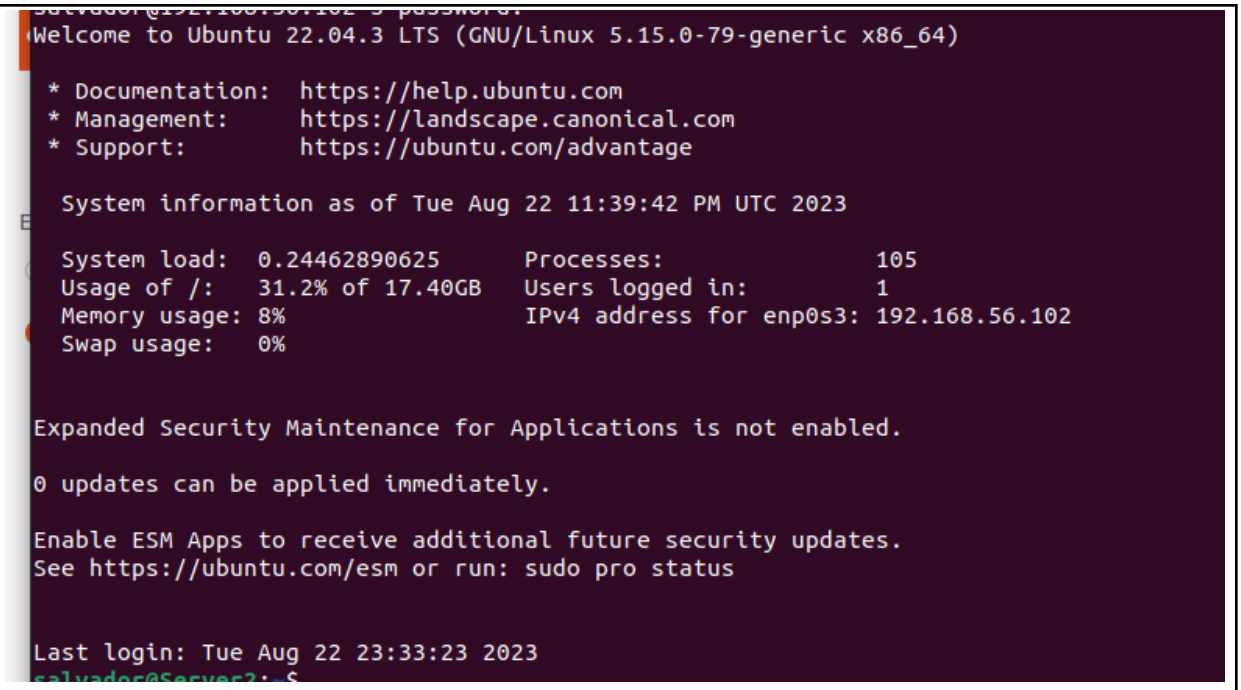
2. Logout of Server 1 by issuing the command *control + D.*

```
salvador@Server1:~$
logout
Connection to 192.168.56.101 closed.
salvador@Workstation:~$
```

3. Do the same for Server 2.

```
salvador@Workstation:~$ ssh salvador@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:naoSkLgM+Dc2E7E2fhtGvnLM2tm85NWx1j3Q4xvSFac.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.
```

```
salvador@192.168.56.102's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 11:39:42 PM UTC 2023
```

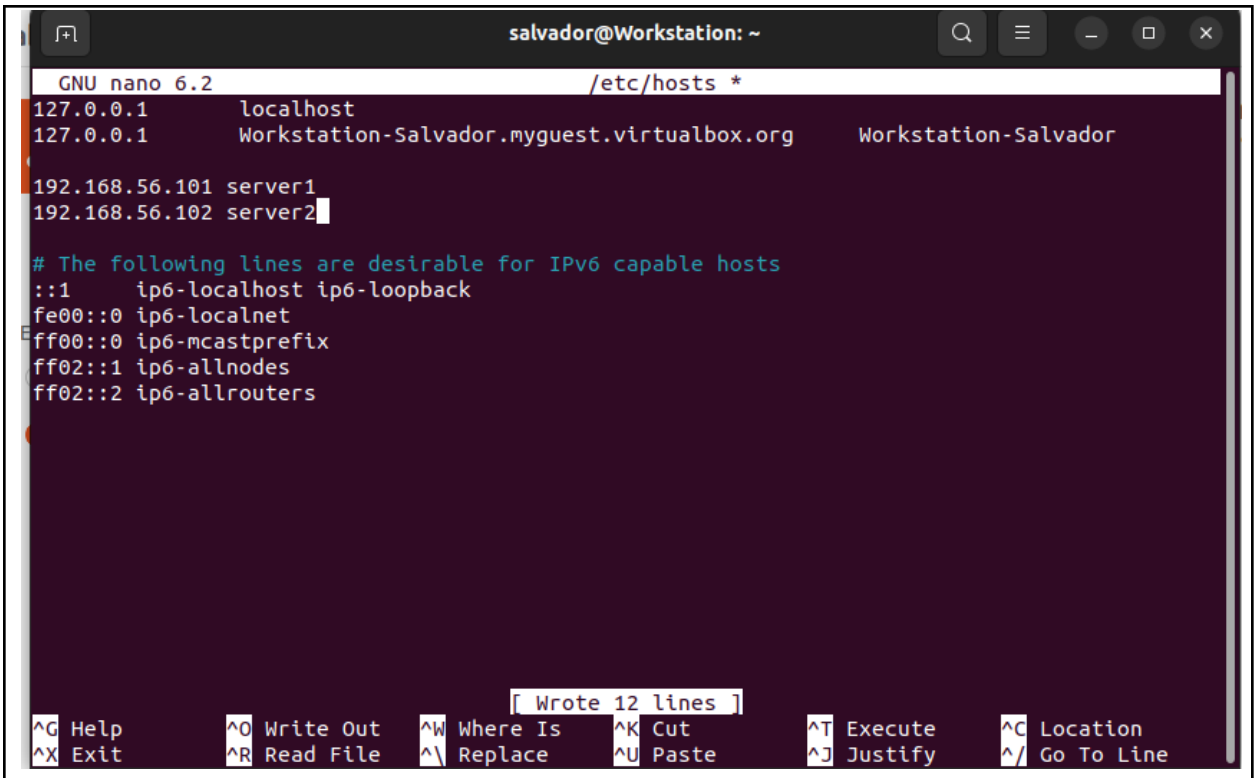


4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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.



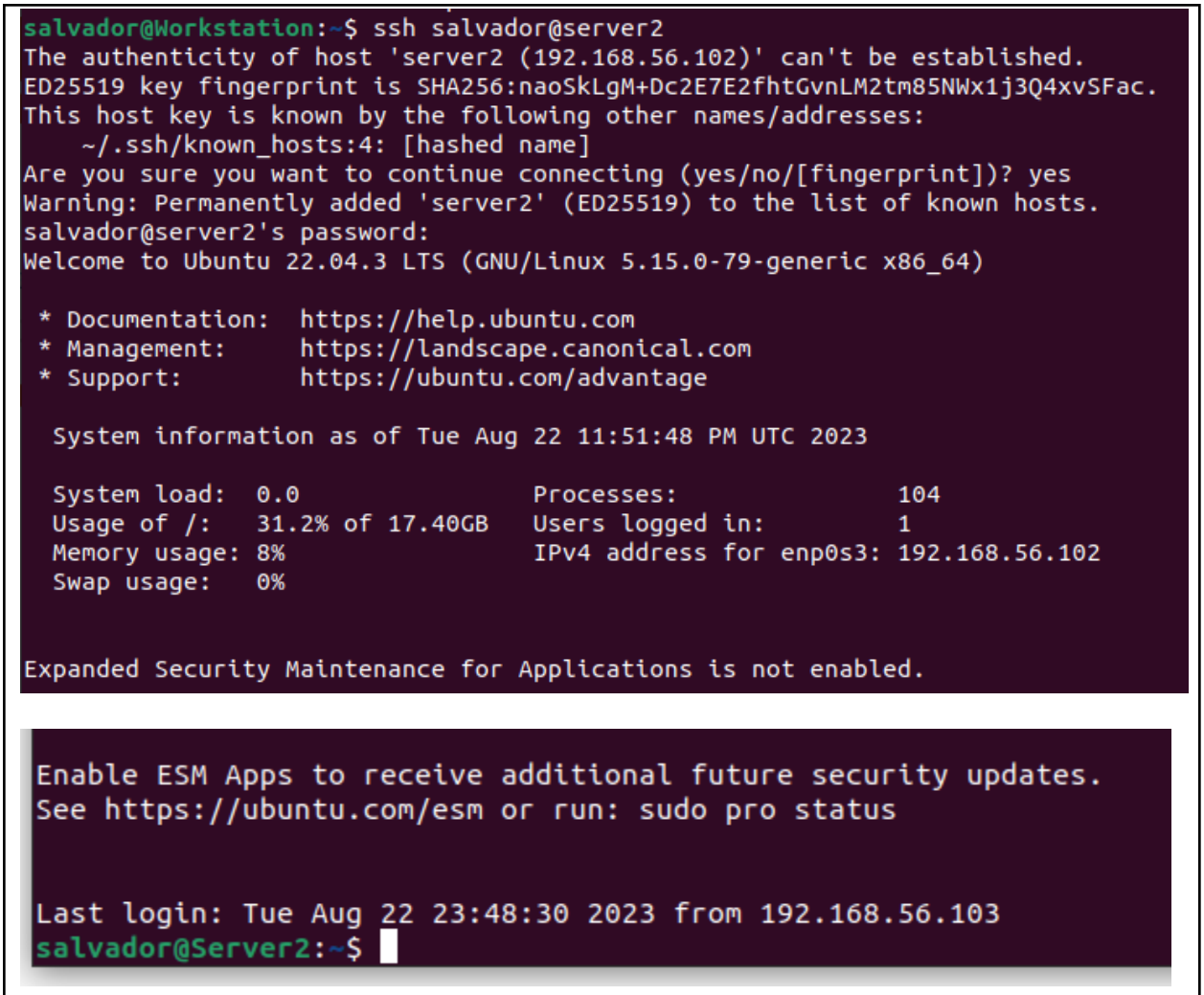
5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

Server 1:

```
salvador@Workstation:~$ ssh salvador@server1
The authenticity of host 'server1 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:3Hfw5vuOnI0D+OpuV+naVNPpUjqNCNVEqZx6Wh08qSQ.
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 'server1' (ED25519) to the list of known hosts.
salvador@server1's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 11:45:58 PM UTC 2023

  System load:  0.0                Processes:               104
  Usage of /:   42.8% of 12.31GB   Users logged in:         1
  Memory usage: 8%                 IPv4 address for enp0s3: 192.168.56.101
  Swap usage:   0%


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


Last login: Tue Aug 22 23:42:03 2023 from 192.168.56.103
salvador@Server1:~$
```

Server 2:



Reflections:

Answer the following:

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

Domain Name System or much more known as DNS is the one responsible for providing us the human-readable hostname for what we need to access. it translates the ip address which we call earlier on “192.168.56.101” for server 1, now upon editing the ssh file we did put server1 along with its ip address which allows us to call the hostname itself instead of the ip address.

2. How secured is SSH?

SSH or Secure shell uses encryption in order to secure the connection between the host and the server, with this they are able to provide a safe space for transaction or transfer of data. The traffic is encrypted, the connections are private because of this

encryption that is happening, browsing or just transferring of files is safe and privately occurring because of the encryption.