Name: Zamora, Angelo E.	**Date Performed:** 08 - 24 - 2024
Course/Section: CPE31S2	**Date Submitted:** 08 - 25 - 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem 2024 - 2025

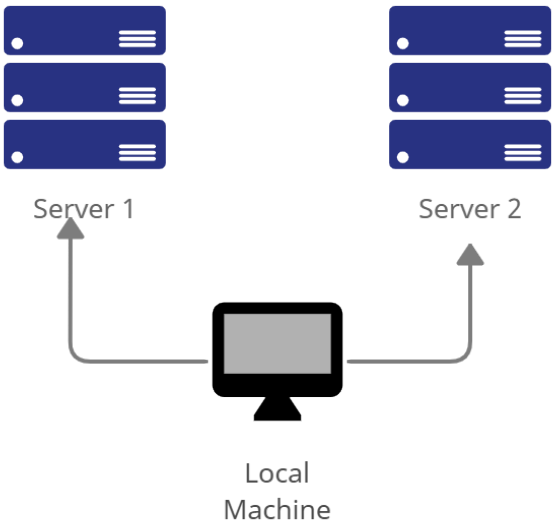
Activity 1: Configure Network using Virtual Machines

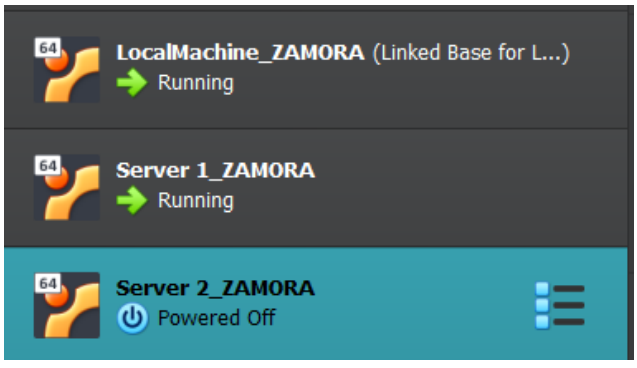
1. **Objectives:**
   1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox
   1.2. Set-up a Virtual Network and Test Connectivity of VMs

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*).

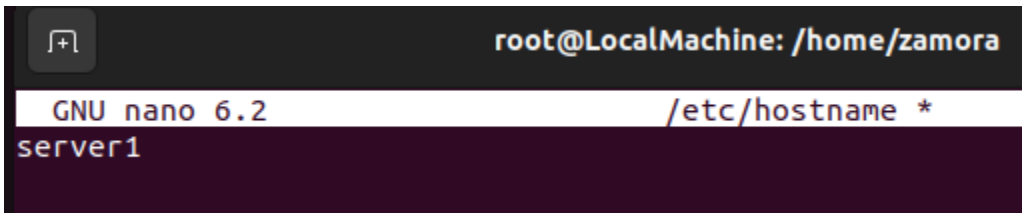


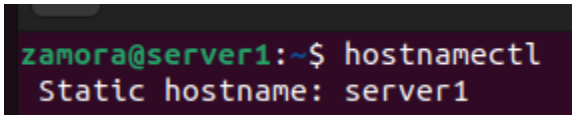


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

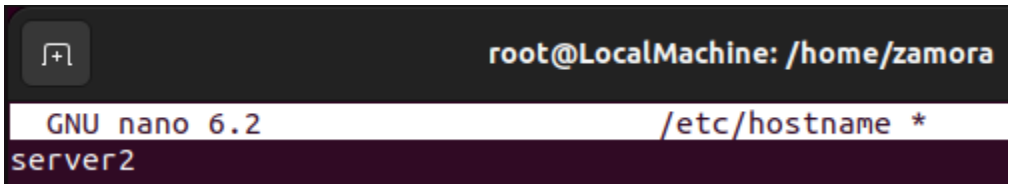
1. Change the hostname using the command *sudo nano /etc/hostname*

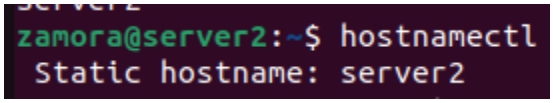
1.1 Use server1 for Server 1



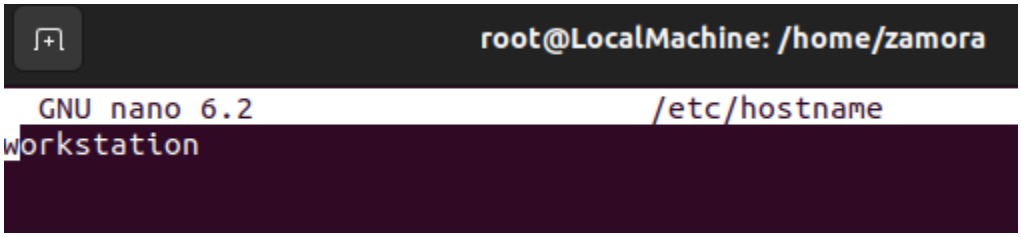


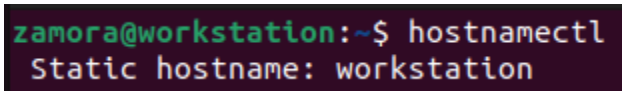
1.2 Use server2 for Server 2



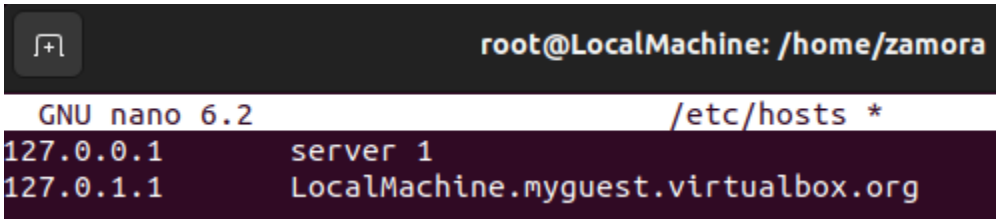


1.3 Use workstation for the Local Machine

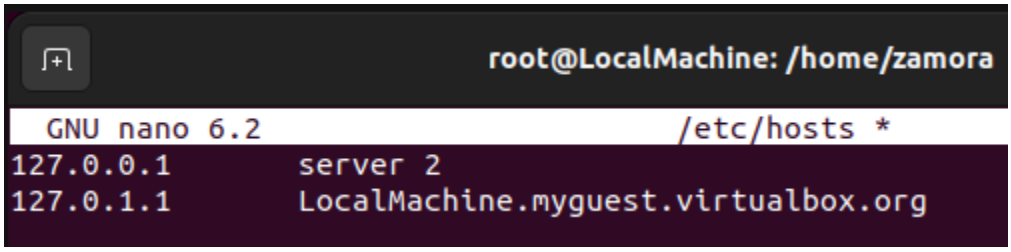




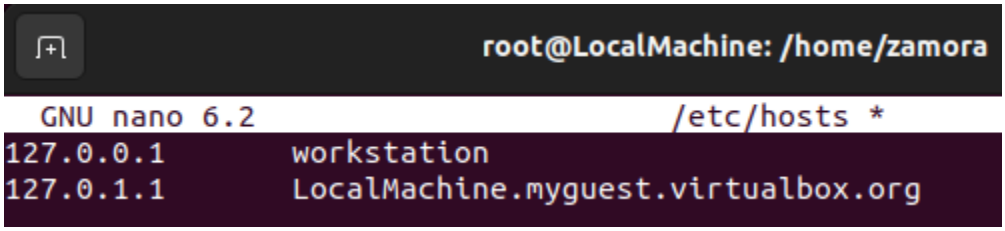
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1



   2.2 Type 127.0.0.1 server 2 for Server 2



   2.3 Type 127.0.0.1 workstation for the Local Machine



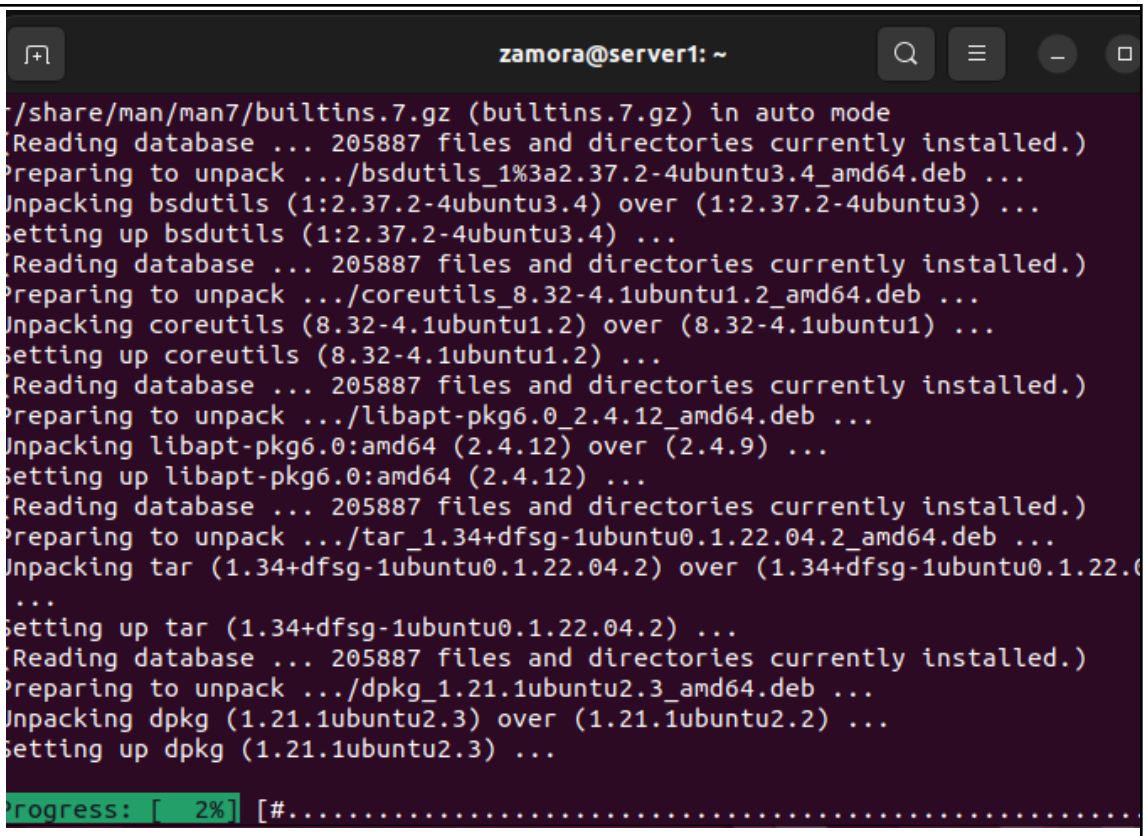
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

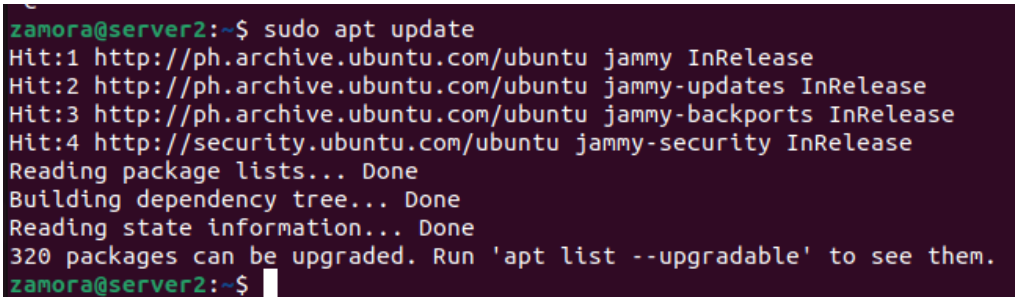
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.

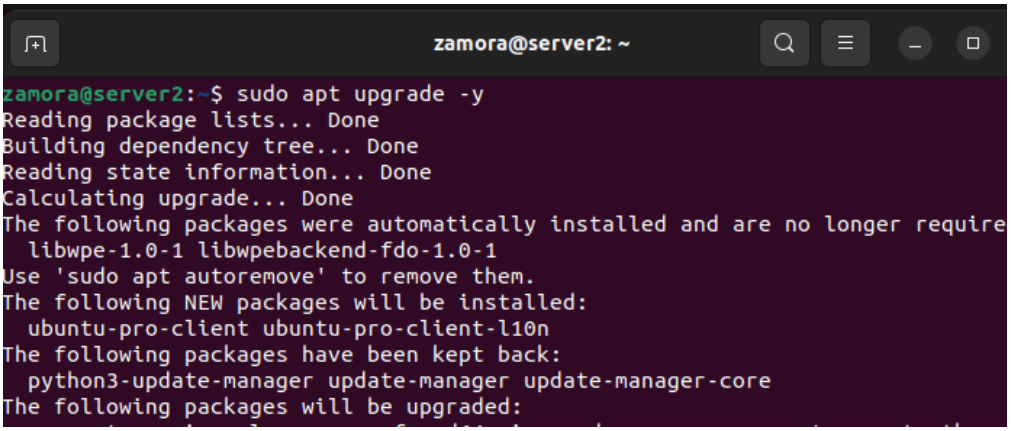
```
zamora@workstation:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
320 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
zamora@workstation:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ubuntu-pro-client ubuntu-pro-client-l10n
The following packages have been kept back:
  python3-update-manager update-manager update-manager-core
The following packages will be upgraded:
  accountsservice alsa-ucm-conf amd64-microcode apparmor apport apport-gtk apt
```

```
zamora@server1:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
320 packages can be upgraded. Run 'apt list --upgradable' to see them.
zamora@server1:~$
```


zamora@server1: ~
/share/man/man7/builtins.7.gz (builtins.7.gz) in auto mode
(Reading database ... 205887 files and directories currently installed.)
Preparing to unpack .../bsdutils_1%3a2.37.2-4ubuntu3.4_amd64.deb ...
Unpacking bsdutils (1:2.37.2-4ubuntu3.4) over (1:2.37.2-4ubuntu3) ...
Setting up bsdutils (1:2.37.2-4ubuntu3.4) ...
(Reading database ... 205887 files and directories currently installed.)
Preparing to unpack .../coreutils_8.32-4.1ubuntu1.2_amd64.deb ...
Unpacking coreutils (8.32-4.1ubuntu1.2) over (8.32-4.1ubuntu1) ...
Setting up coreutils (8.32-4.1ubuntu1.2) ...
(Reading database ... 205887 files and directories currently installed.)
Preparing to unpack .../libapt-pkg6.0_2.4.12_amd64.deb ...
Unpacking libapt-pkg6.0:amd64 (2.4.12) over (2.4.9) ...
Setting up libapt-pkg6.0:amd64 (2.4.12) ...
(Reading database ... 205887 files and directories currently installed.)
Preparing to unpack .../tar_1.34+dfsg-1ubuntu0.1.22.04.2_amd64.deb ...
Unpacking tar (1.34+dfsg-1ubuntu0.1.22.04.2) over (1.34+dfsg-1ubuntu0.1.22.
...
Setting up tar (1.34+dfsg-1ubuntu0.1.22.04.2) ...
(Reading database ... 205887 files and directories currently installed.)
Preparing to unpack .../dpkg_1.21.1ubuntu2.3_amd64.deb ...
Unpacking dpkg (1.21.1ubuntu2.3) over (1.21.1ubuntu2.2) ...
Setting up dpkg (1.21.1ubuntu2.3) ...
Progress: [2%] [#.....


zamora@server2:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
320 packages can be upgraded. Run 'apt list --upgradable' to see them.
zamora@server2:~$


zamora@server2: ~
zamora@server2:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer require
libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
ubuntu-pro-client ubuntu-pro-client-l10n
The following packages have been kept back:
python3-update-manager update-manager update-manager-core
The following packages will be upgraded:

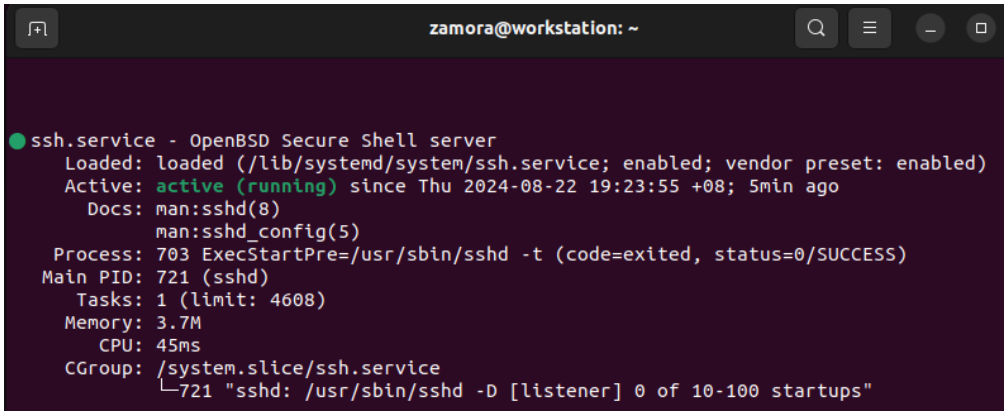
2. Install the SSH server using the command *sudo apt install openssh-server*.

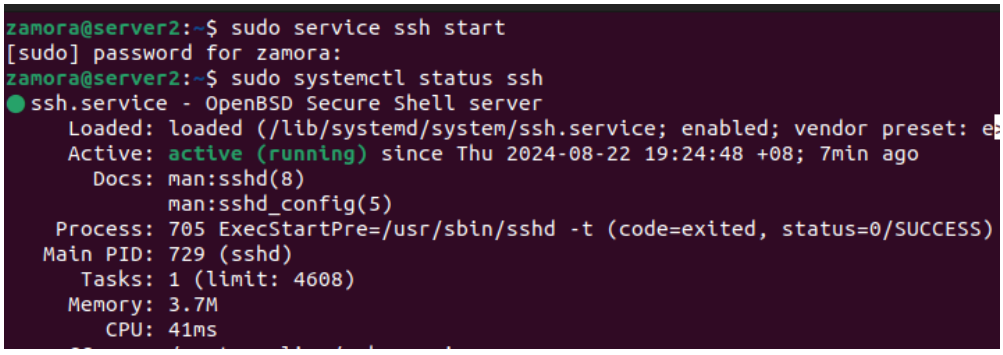
```
zamora@workstation:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
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 3 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1
Fetched 751 kB in 1s (1,454 kB/s)
```

```
zamora@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
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 3 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-
server amd64 1:8.9p1-3ubuntu0.10 [38.9 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-serve
r amd64 1:8.9p1-3ubuntu0.10 [435 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term
all 6.3-2ubuntu0.1 [267 kB]
```

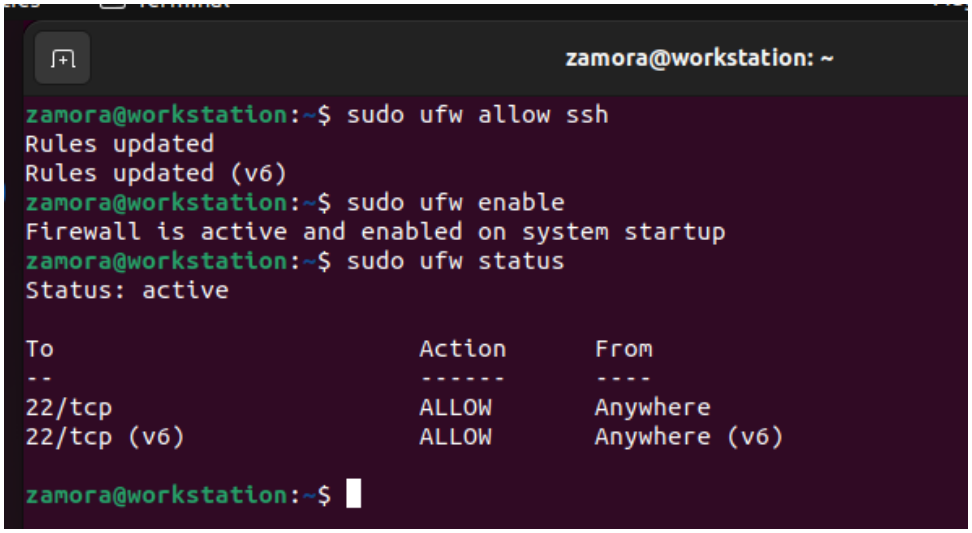
```
zamora@server2:~$ sudo apt install openssh-server -y
[sudo] password for zamora:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
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
```

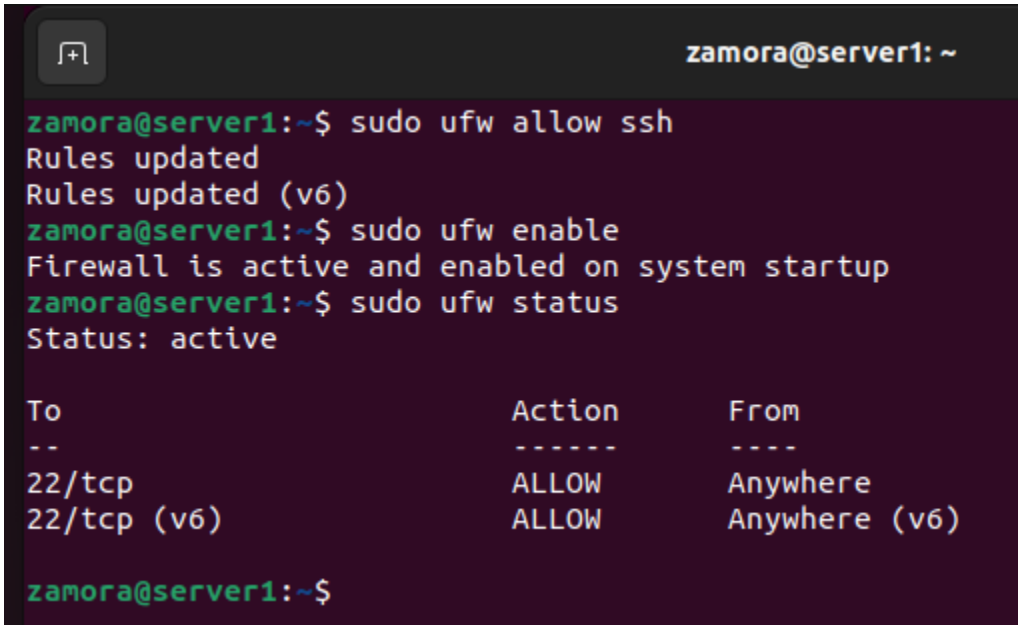
3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*

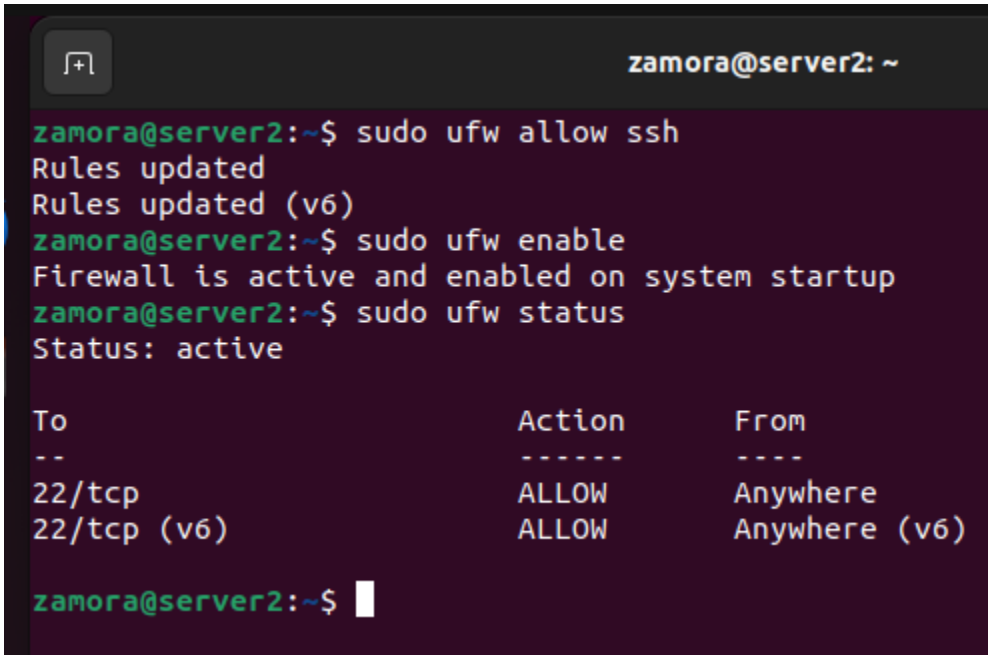




4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*

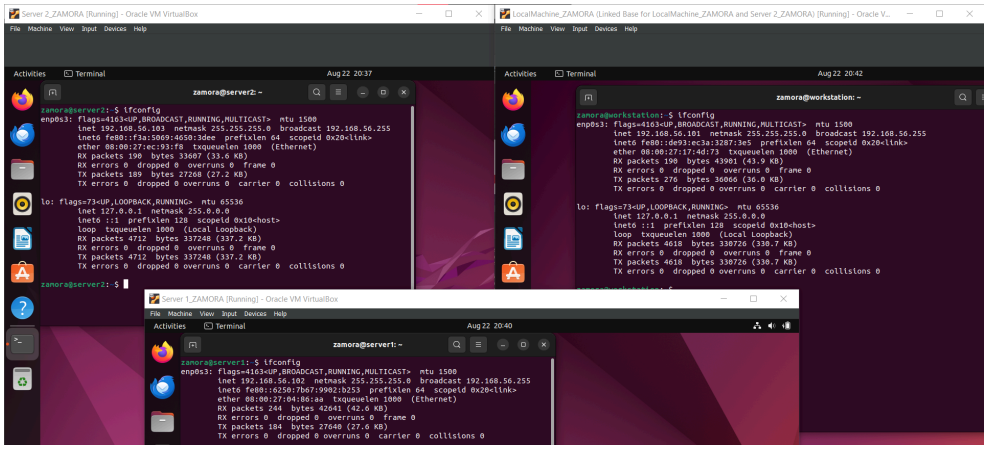





zamora@server2: ~
zamora@server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
zamora@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
zamora@server2:~$ sudo ufw status
Status: active
To Action From
-- ------ ----
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
zamora@server2:~$

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.**102**
   1.2 Server 2 IP address: 192.168.56.**103**
   1.3 Workstation IP address: 192.168.56.**101**



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

```
zamora@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=0.436 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.286 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.288 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.283 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.286 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.279 ms
^C
--- 192.168.56.102 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5112ms
rtt min/avg/max/mdev = 0.279/0.309/0.436/0.056 ms
zamora@workstation:~$
```

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

```
zamora@workstation:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.581 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.271 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.271 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.288 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.273 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.270 ms
^C
--- 192.168.56.103 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5116ms
rtt min/avg/max/mdev = 0.270/0.325/0.581/0.114 ms
zamora@workstation:~$
```

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

```
zamora@server1:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.479 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.285 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.272 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.278 ms
^C
--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.268/0.316/0.479/0.081 ms
zamora@server1:~$
```

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*

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

3. Do the same for Server 2.

```
zamora@workstation:~$ ssh zamora@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:HgNY0KuNeiVz8/JLssLzX3d1CgUFW45svI+epp6N1sQ.
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 '192.168.56.103' (ED25519) to the list of known hosts.
zamora@192.168.56.103's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.8.0-40-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.

zamora@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)

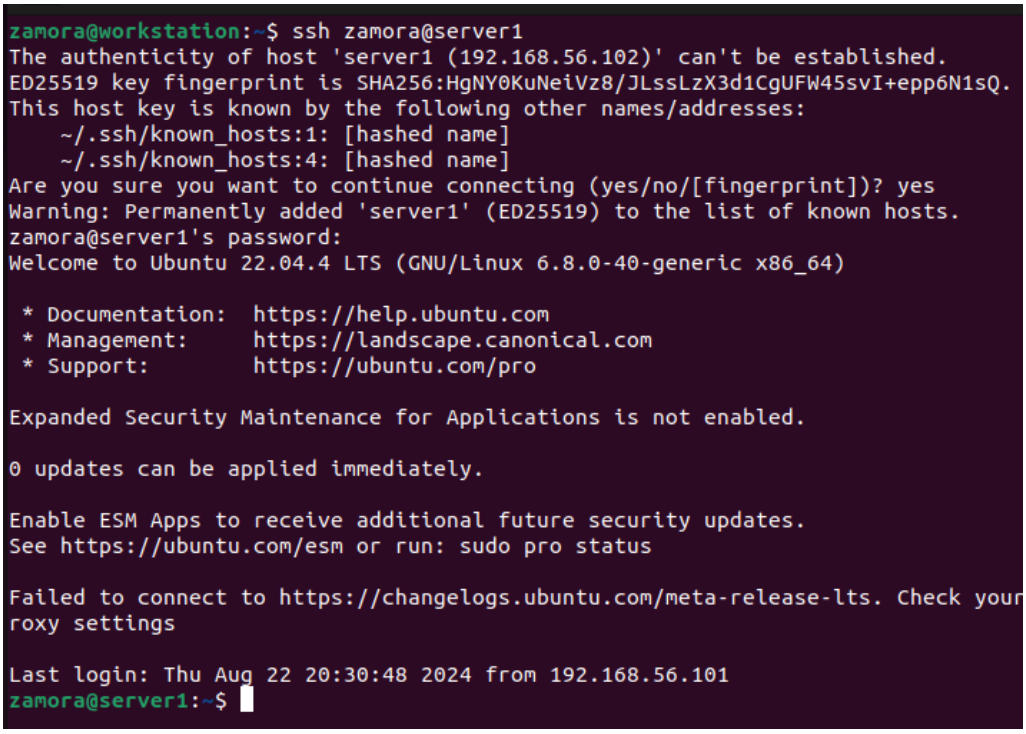
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

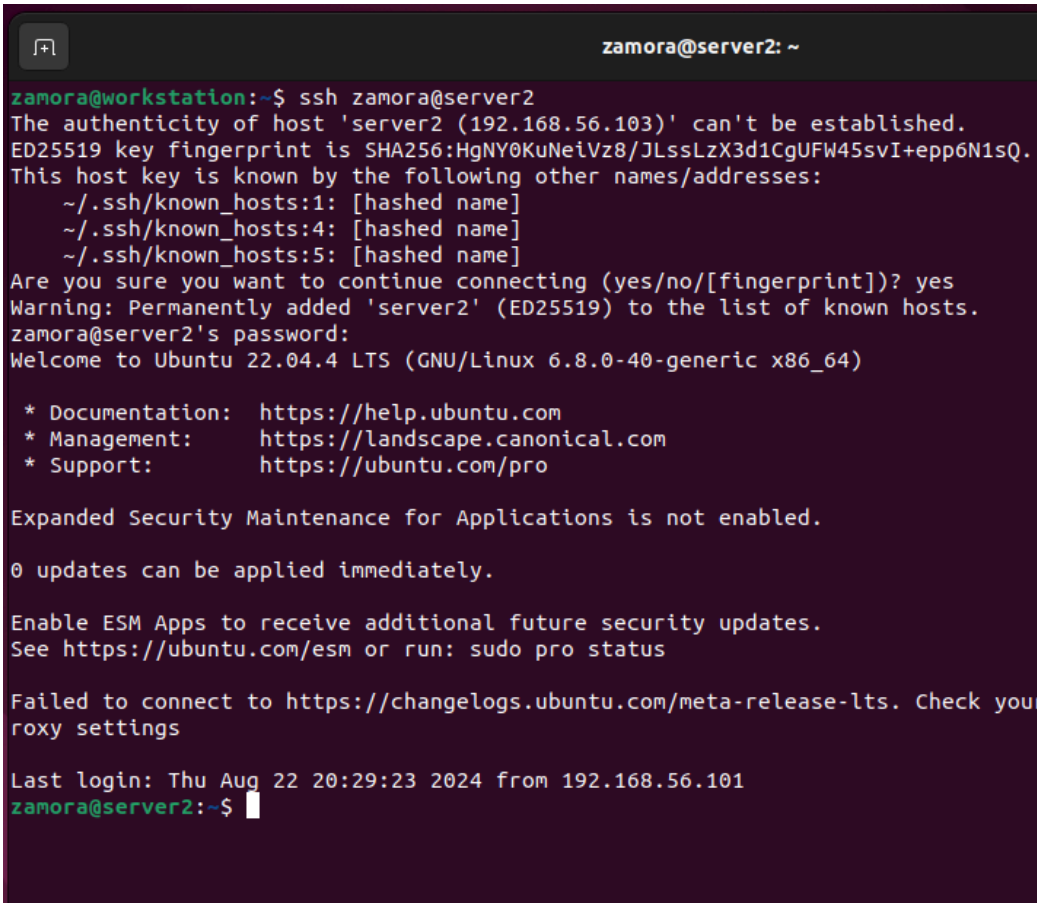
```
                                zamora@workstation: ~
  GNU nano 6.2                                /etc/hosts
127.0.0.1       workstation
192.168.56.102  server1
192.168.56.103  server2
127.0.1.1       LocalMachine.myguest.virtualbox.org     LocalMachine
```

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.

   SSH to Server 1



SSH to Server 2



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   When importing another Linux machine's IP address and hostname, the IP address must come first, followed by the hostname. **(eg. 192.168.102 server1)**

2. How secured is SSH?
   SSH is a secure mechanism for remote administration and communication in Linux. It has robust authentication techniques and encryptions to prevent unauthorized access while also providing strong security.