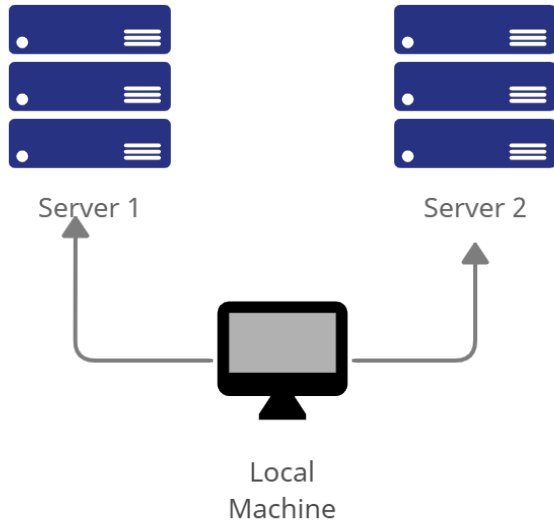
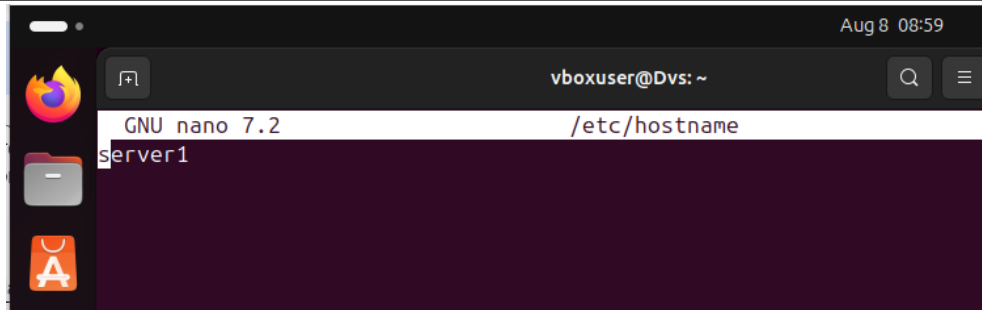
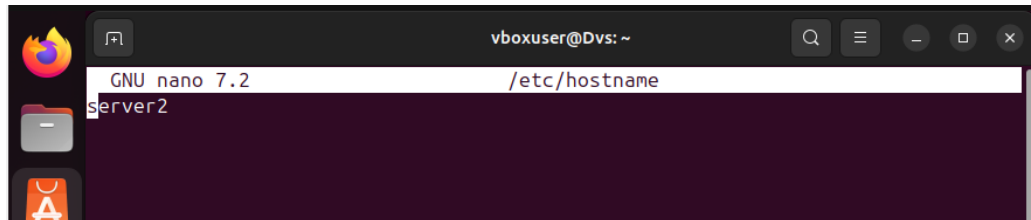


Name: Santos, Emmanuelle Dave G.	Date Performed: 08/08/2025
Course/Section:CPE32S2	Date Submitted:08/08/2025
Instructor: Engr. Robin Valenzuela	Semester and SY: 2025-2026
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft Azure 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, <i>provide screenshots for each task.</i> (Note: <i>it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine</i>).	
 <pre> graph TD LocalMachine[Local Machine] --> Server1[Server 1] LocalMachine --> Server2[Server 2] </pre> <p>The diagram illustrates a network topology where a central 'Local Machine' (represented by a monitor icon) is connected to two separate server stacks. 'Server 1' and 'Server 2' are each represented by three stacked server rack icons. Arrows point from the 'Local Machine' to both 'Server 1' and 'Server 2', indicating network connectivity.</p>	
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. <ol style="list-style-type: none"> Change the hostname using the command <i>sudo nano /etc/hostname</i> <ol style="list-style-type: none"> Use server1 for Server 1 	



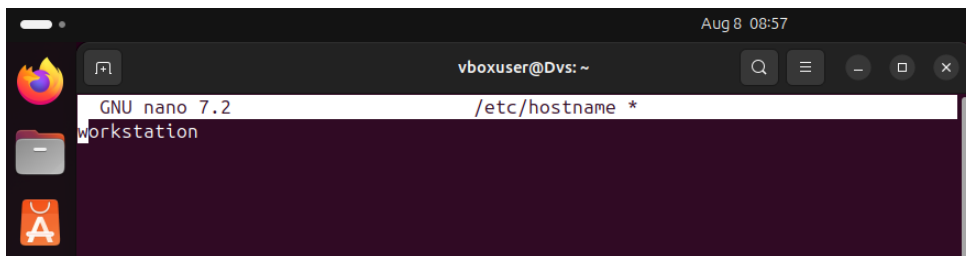
A terminal window titled 'vboxuser@Dvs: ~' with a search bar and menu icon. It shows the GNU nano 7.2 editor editing the file /etc/hostname. The content of the file is 'Server1'.

1.2 Use server2 for Server 2



A terminal window titled 'vboxuser@Dvs: ~' with a search bar and menu icon. It shows the GNU nano 7.2 editor editing the file /etc/hostname. The content of the file is 'server2'.

1.3 Use workstation for the Local Machine

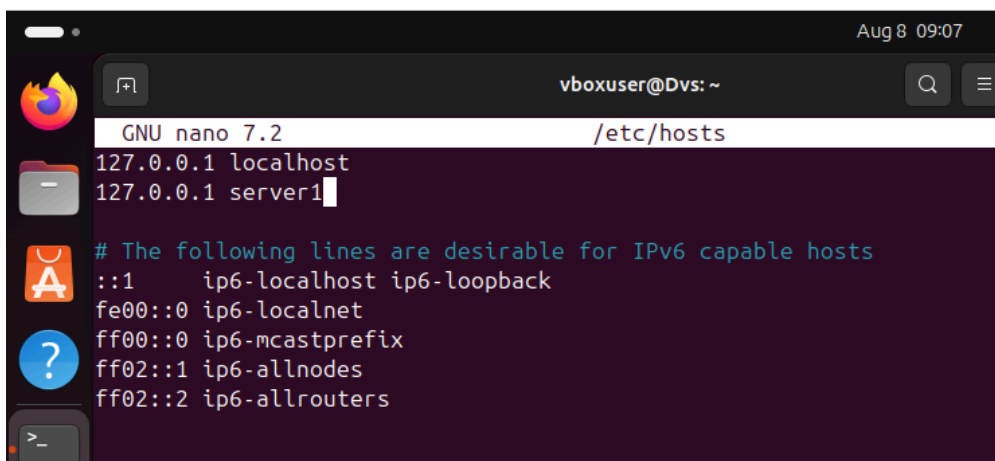


A terminal window titled 'vboxuser@Dvs: ~' with a search bar and menu icon. It shows the GNU nano 7.2 editor editing the file /etc/hostname *. The content of the file is 'workstation'.

1.4

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



A terminal window titled 'vboxuser@Dvs: ~' with a search bar and menu icon. It shows the GNU nano 7.2 editor editing the file /etc/hosts. The content of the file is:

```
127.0.0.1 localhost
127.0.0.1 server1

# 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 localhost
127.0.0.1 server2

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

2.3 Type 127.0.0.1 workstation for the Local Machine

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

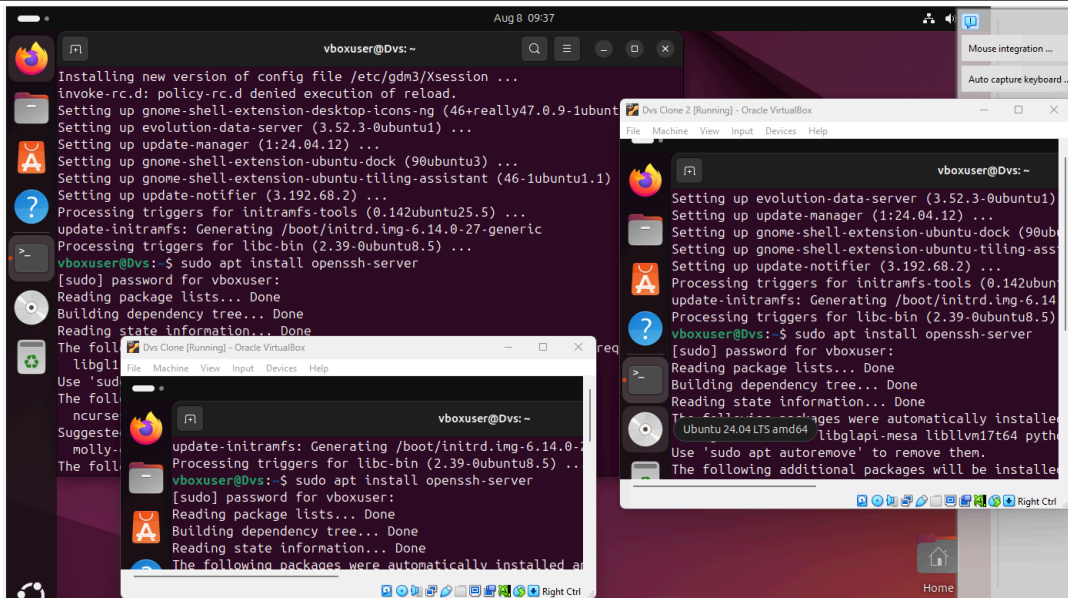
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

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.

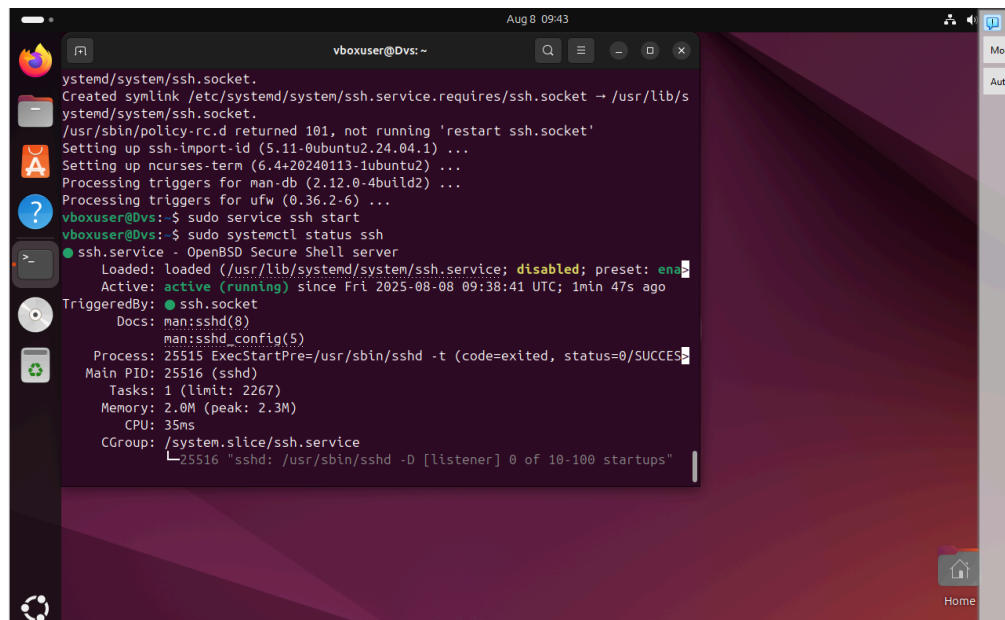
```
Get:39 http://ph.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Comp
onents [212 B]
Reading package lists... Done
E: The repository 'file:/cdrom noble
N: Updating from such a repository c
bled by default.
N: See apt-secure(8) manpage for re
ils.
vboxuser@Dvs:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automati
libgl1-amd64 libglapi-mesa lib
Use 'sudo apt autoremove' to remove
The following NEW packages will be
libllv19 libmalcontent-0-0 mesa-
The following packages have been ke
libgl1-amd64 libglapi-mesa lib
The following packages will be upgr
acl alsa-ucm-conf apparmor apt ap
bluez-obexd bsdextrautils bsdutil
```

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

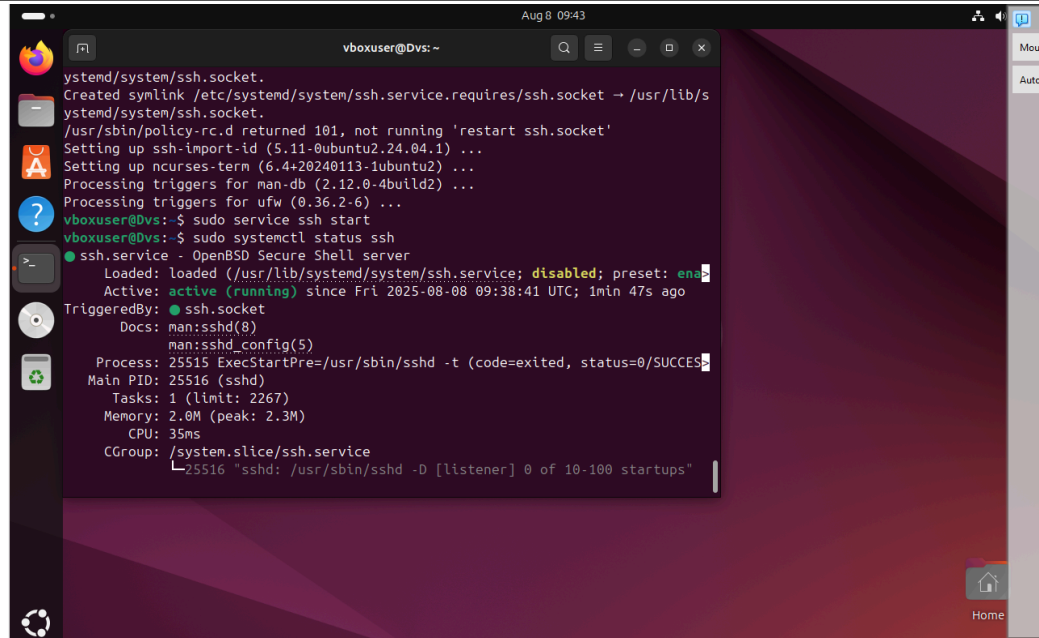


4. Verify if the SSH service has started by issuing the following commands:

3.1 *sudo service ssh start*



3.2 *sudo systemctl status ssh*



```
vboxuser@Dvs: ~  
systemd/system/ssh.socket.  
Created symlink /etc/systemd/system/ssh.service.requires/ssh.socket → /usr/lib/s  
systemd/system/ssh.socket.  
/usr/sbin/policy-rc.d returned 101, not running 'restart ssh.socket'  
Setting up ssh-import-id (5.11-0ubuntu2.24.04.1) ...  
Setting up ncurses-term (6.4+20240113-1ubuntu2) ...  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for ufw (0.36.2-6) ...  
vboxuser@Dvs:~$ sudo service ssh start  
vboxuser@Dvs:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena  
Active: active (running) since Fri 2025-08-08 09:38:41 UTC; 1min 47s ago  
TriggeredBy: ● ssh.socket  
Docs: man:sshd(8)  
man:sshd_config(5)  
Process: 25515 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
Main PID: 25516 (sshd)  
Tasks: 1 (limit: 2267)  
Memory: 2.0M (peak: 2.3M)  
CPU: 35ms  
CGroup: /system.slice/ssh.service  
└─25516 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

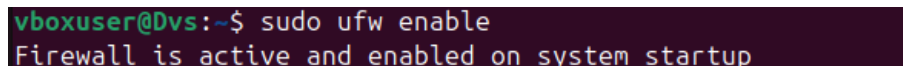
5. Configure the firewall to all port 22 by issuing the following commands:

4.1 *sudo ufw allow ssh*



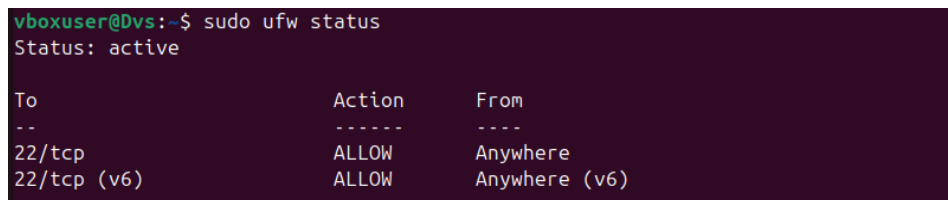
```
vboxuser@Dvs:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)  
vboxuser@Dvs:~$
```

4.2 *sudo ufw enable*



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

4.3 *sudo ufw status*



```
vboxuser@Dvs:~$ 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: 192.168.56.102
 - 1.2 Server 2 IP address: 192.168.56.103
 - 1.3 Server 3 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

```
vboxuser@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=2.59 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.11 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=1.08 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.942 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.538 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.758 ms
2.2 64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.529 ms
```

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

```
vboxuser@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=6.70 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.899 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.520 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.09 ms
^C
```


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

```
vboxuser@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=1.64 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.871 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.823 ms
^C
--- 192.168.56.103 ping statistics ---
```

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 jvtaylor@192.168.56.120`



```

rtt min/avg/max/mdev = 0.520/2.303/6.703/2.548 ms
vboxuser@workstation:~$ ssh vboxuser@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be
ED25519 key fingerprint is SHA256:jFRhcr690KzO92CxEub7x59hPqsQ5wdU
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])
Warning: Permanently added '192.168.56.102' (ED25519) to the list of
known hosts.
vboxuser@192.168.56.102's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

```



- 1.2 Enter the password for server 1 when prompted

```


vboxuser@192.168.56.102's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

```


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



```

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.

vboxuser@server1:~$

```

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

```

vboxuser@server1:~$
logout
Connection to 192.168.56.102 closed.
vboxuser@workstation:~$

```

3. Do the same for Server 2.

```

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted
by applicable law.

vboxuser@server2:~$

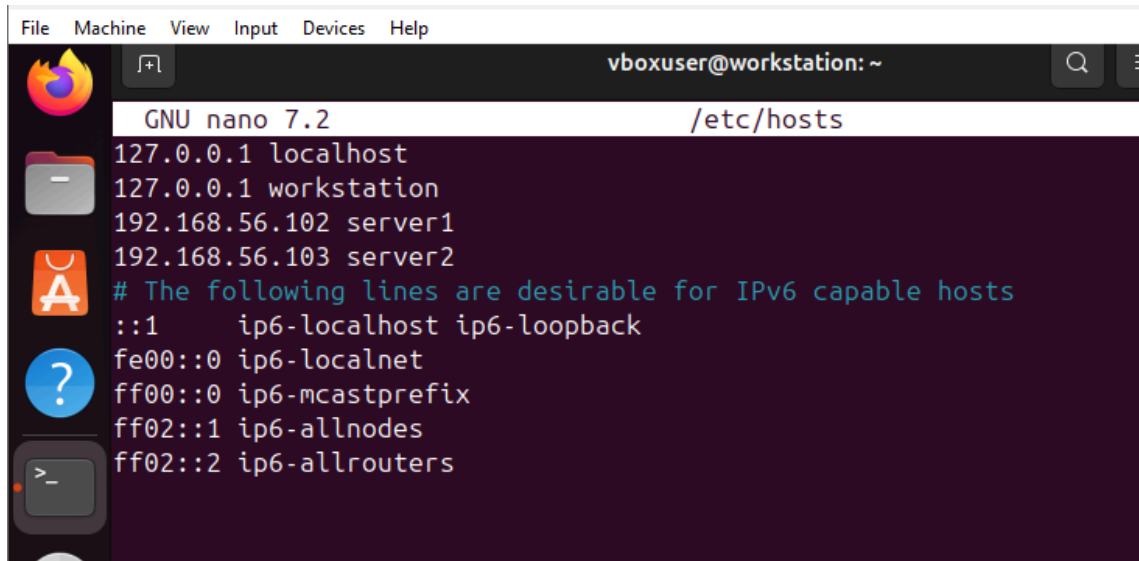
```



```
vboxuser@server2:~$  
logout  
Connection to 192.168.56.103 closed.  
vboxuser@workstation:~$
```

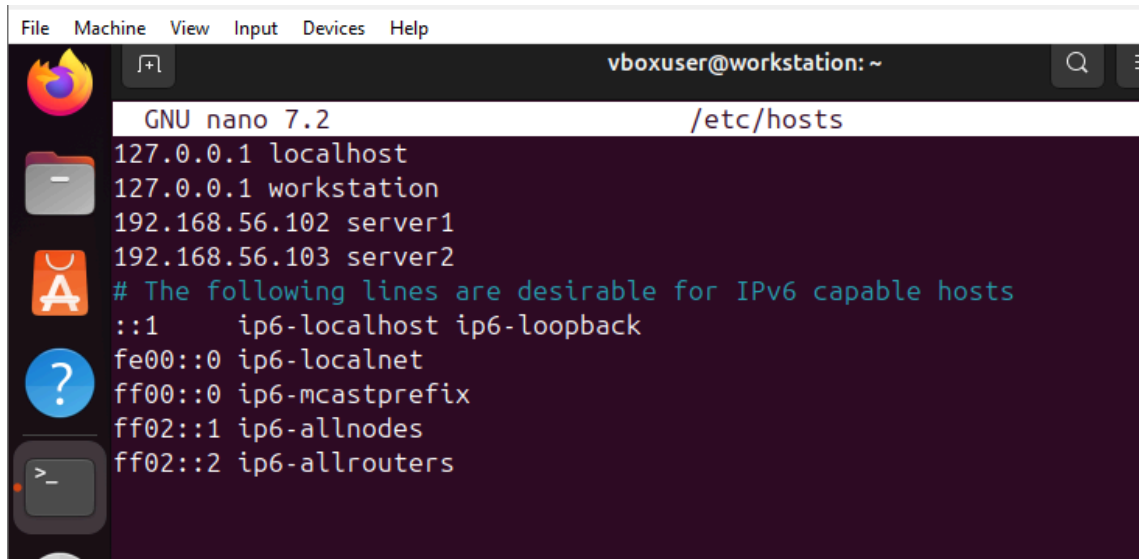
4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts*. Below all texts type the following:

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



```
File Machine View Input Devices Help  
vboxuser@workstation: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1 localhost  
127.0.0.1 workstation  
192.168.56.102 server1  
192.168.56.103 server2  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

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



```
File Machine View Input Devices Help  
vboxuser@workstation: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1 localhost  
127.0.0.1 workstation  
192.168.56.102 server1  
192.168.56.103 server2  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

4.3 Save the file and exit.


```
File Machine View Input Devices Help
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
vboxuser@server2:~$
logout
Connection to 192.168.56.103 closed.
vboxuser@workstation:~$ sudo nano /etc/hosts
[sudo] password for vboxuser:
vboxuser@workstation:~$
```

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 jvtaylor@server1`. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

```
vboxuser@workstation:~$ ssh vboxuser@server1
vboxuser@server1's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Fri Aug  8 10:21:29 2025 from 192.168.56.101
vboxuser@server1:~$ ssh vboxuser@server2
ssh: Could not resolve hostname server2: Temporary failure in name resolution
vboxuser@server1:~$
logout
Connection to server1 closed.
```

```
ED25519 key fingerprint is SHA256:jFRhcr690Kz092CxEub7x59hPqsQ5wdUFeEBxKT3
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2' (ED25519) to the list of known hosts.
vboxuser@server2's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

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

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

Last login: Fri Aug  8 10:26:43 2025 from 192.168.56.101
vboxuser@server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
because the IP address is local to the the servers in can be used instead of names
2. How secured is SSH?

SSH is secured because it always required a password if you create a user ssh and cannot not easily bybass