Name: Christian Dave L. Berja	**Date Performed: 22/08/2023**
Course/Section: CPE31S5	**Date Submitted:23/08/2023**
Instructor: Prof. Roman	**Semester and SY: 1st semester 3rd yr.**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

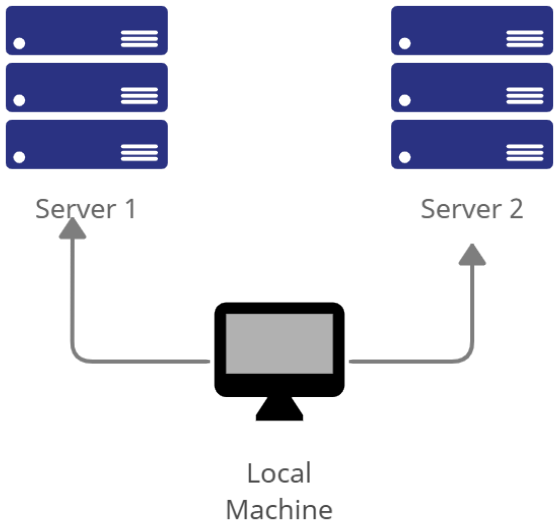
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

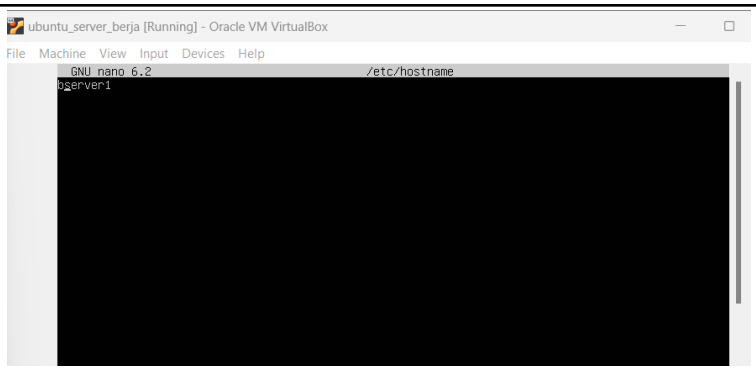
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



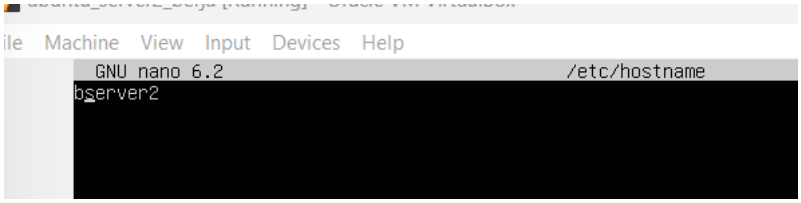
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

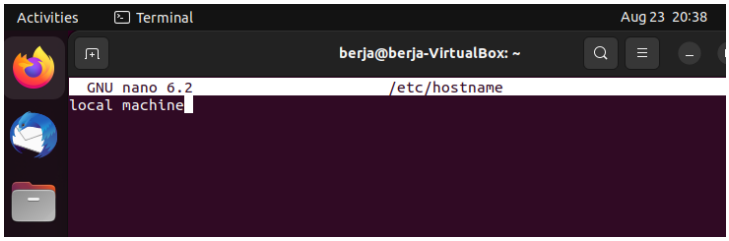
1.1 Use server1 for Server 1



1.2 Use server2 for Server 2

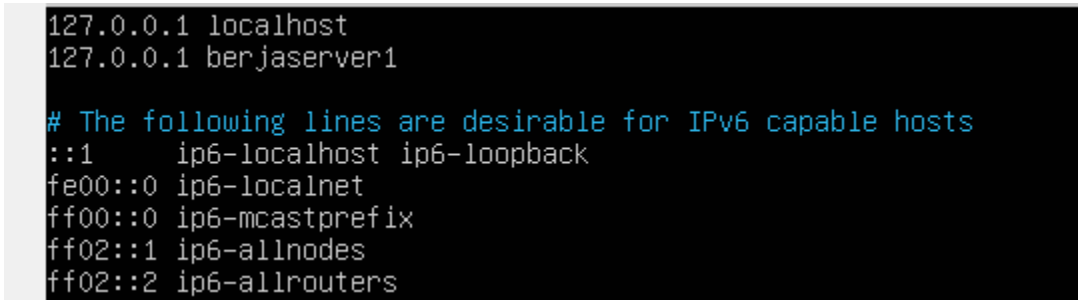


1.3 Use workstation for the Local Machine

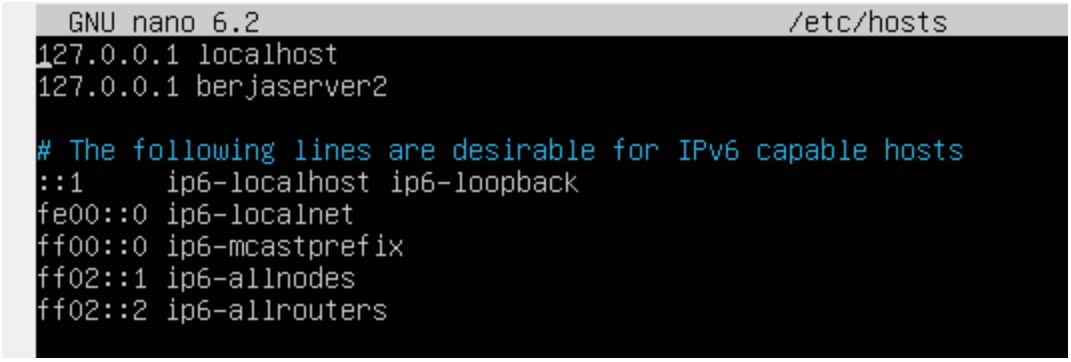


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

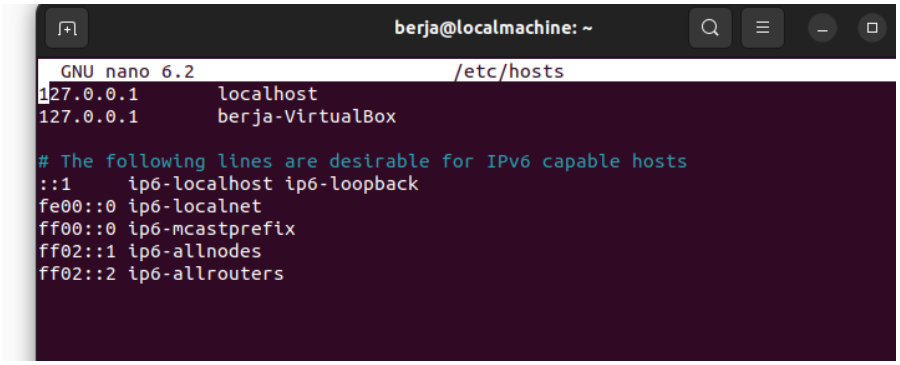
2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2

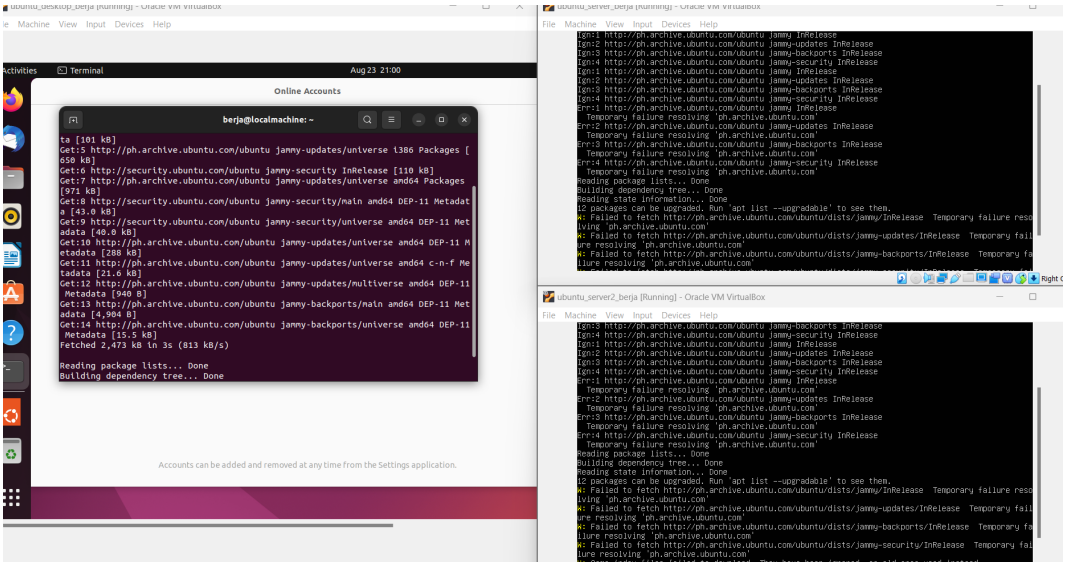


2.3 Type 127.0.0.1 workstation for the Local Machine



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.



```
tting up libtiff5:amd64 (4.3.0-6ubuntu0.5) ...
tting up initramfs-tools-bin (0.140ubuntu13.4) ...
tting up gir1.2-webkit2-4.0:amd64 (2.40.5-0ubuntu0.22.04.1) ...
tting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.4) ...
tting up samba-libs:amd64 (2:4.15.13+dfsg-0ubuntu1.3) ...
tting up vim-tiny (2:8.2.3995-1ubuntu2.11) ...
tting up ghostscript (9.55.0~dfsg1-0ubuntu5.4) ...
tting up libsmbclient:amd64 (2:4.15.13+dfsg-0ubuntu1.3) ...
tting up initramfs-tools-core (0.140ubuntu13.4) ...
tting up initramfs-tools (0.140ubuntu13.4) ...
date-initramfs: deferring update (trigger activated)
tting up ghostscript-x (9.55.0~dfsg1-0ubuntu5.4) ...
ocessing triggers for mailcap (3.70+nmu1ubuntu1) ...
ocessing triggers for desktop-file-utils (0.26-1ubuntu3) ...
ocessing triggers for hicolor-icon-theme (0.17-2) ...
ocessing triggers for gnome-menus (3.36.0-1ubuntu3) ...
ocessing triggers for libc-bin (2.35-0ubuntu3.1) ...
ocessing triggers for man-db (2.10.2-1) ...
ocessing triggers for initramfs-tools (0.140ubuntu13.4) ...
date-initramfs: Generating /boot/initrd.img-6.2.0-26-generic
rja@localmachine:~$
```

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

```
Setting up openssh-sftp-server (1:8.9p1-3ubuntu0.3) ...
Setting up openssh-server (1:8.9p1-3ubuntu0.3) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:cQtRyk1etall700NqAqR+62hwocysTwSK3qvjHa3Urw root@localmac
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:4Vm3pGxiVnBgZDLB9U36Grm7vciHoQOX1/VfEqBK2XI root@localmach
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:8hsy+HPlqShBm5IPR0+r3nL11DpALmFU07Q8h4y5LBw root@localmach
9)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/syste
ce.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.servi
ystemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
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) ...
berja@localmachine:~$
```

```
W: Some index files failed to download. They have been ignored, or old
berja@bserver1:~$ 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.
berja@bserver1:~$
```

```
W: Some index files failed to download. They have been ignored, or old on
berja2@bserver2:~$ 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.
berja2@bserver2:~$
```

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

3.1 *sudo service ssh start*

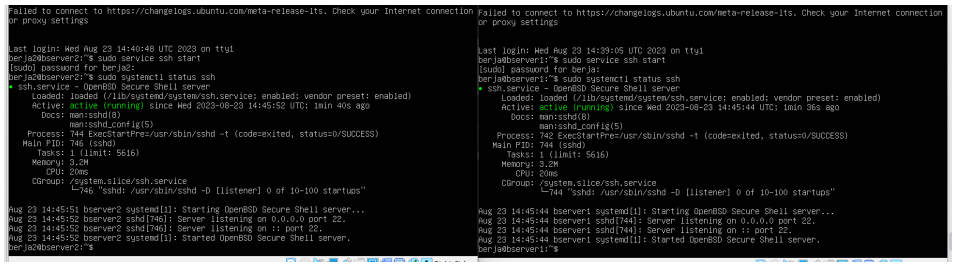
```
berja@localmachine:~$ sudo service ssh start
[sudo] password for berja:
Job for ssh.service failed because the control process exited with error code.
See "systemctl status ssh.service" and "journalctl -xeu ssh.service" for details
.
```

```
Last login: Wed Aug 23 14:39:05 UTC 2023 on tty1
berja@bserver1:~$ sudo service ssh start
[sudo] password for berja:
berja@bserver1:~$
```

```
Last login: Wed Aug 23 14:40:48 UTC 2023 on tty1
berja2@bserver2:~$ sudo service ssh start
[sudo] password for berja2:
berja2@bserver2:~$ _
```

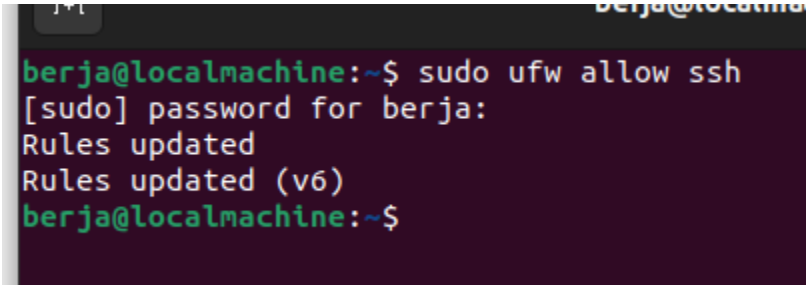
3.2 *sudo systemctl status ssh*

```
.
berja@localmachine:~$ sudo systemctl status ssh
× ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: failed (Result: exit-code) since Wed 2023-08-23 22:44:08 PST; 47s >
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1957 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=1/FAILURE)
        CPU: 6ms

Aug 23 22:44:08 localmachine systemd[1]: ssh.service: Control process exited, c>
Aug 23 22:44:08 localmachine systemd[1]: ssh.service: Failed with result 'exit->
Aug 23 22:44:08 localmachine systemd[1]: Failed to start OpenBSD Secure Shell s>
Aug 23 22:44:08 localmachine systemd[1]: ssh.service: Scheduled restart job, re>
Aug 23 22:44:08 localmachine systemd[1]: Stopped OpenBSD Secure Shell server.
Aug 23 22:44:08 localmachine systemd[1]: ssh.service: Start request repeated to>
Aug 23 22:44:08 localmachine systemd[1]: ssh.service: Failed with result 'exit->
Aug 23 22:44:08 localmachine systemd[1]: Failed to start OpenBSD Secure Shell s>
lines 1-16/16 (END)
```

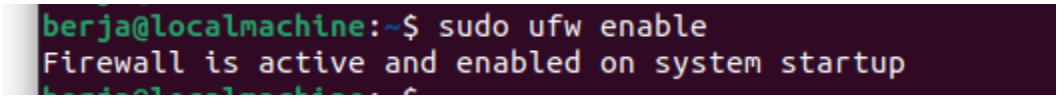


4. Configure the firewall to all port 22 by issuing the following commands:

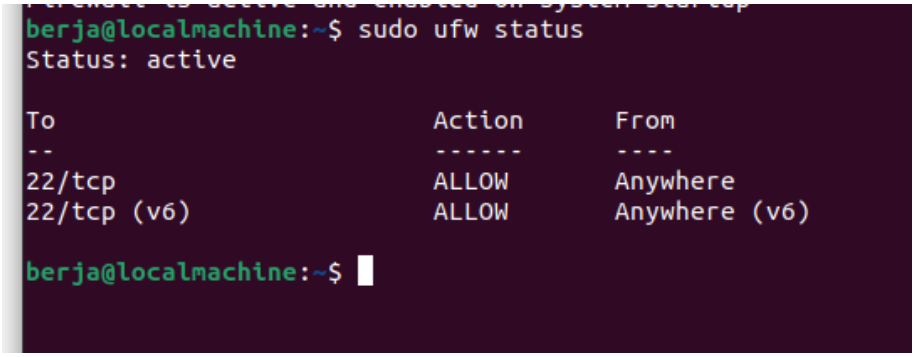
4.1 *sudo ufw allow ssh*

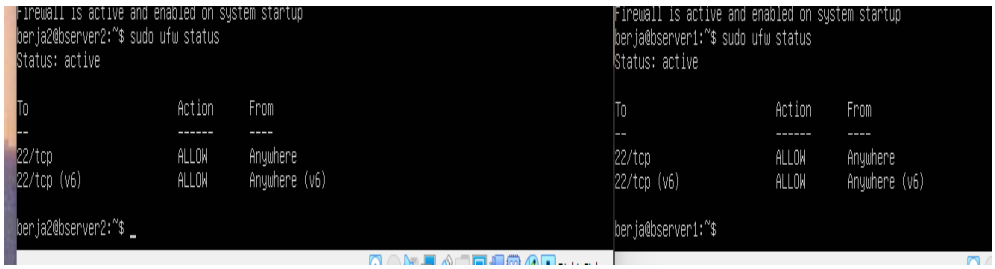


4.2 *sudo ufw enable*



4.3 *sudo ufw status*





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: 192.168.56.109
   1.3 Server 3 IP address: 192.168.56.110
2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ■ Successful ☐ Not Successful

```
berja@localmachine:~$ ping 192.168.56.109
PING 192.168.56.109 (192.168.56.109) 56(84) bytes of data.
64 bytes from 192.168.56.109: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.56.109: icmp_seq=2 ttl=64 time=0.329 ms
64 bytes from 192.168.56.109: icmp_seq=3 ttl=64 time=0.372 ms
64 bytes from 192.168.56.109: icmp_seq=4 ttl=64 time=0.310 ms
64 bytes from 192.168.56.109: icmp_seq=5 ttl=64 time=0.358 ms
64 bytes from 192.168.56.109: icmp_seq=6 ttl=64 time=0.354 ms
64 bytes from 192.168.56.109: icmp_seq=7 ttl=64 time=0.206 ms
64 bytes from 192.168.56.109: icmp_seq=8 ttl=64 time=0.203 ms
64 bytes from 192.168.56.109: icmp_seq=9 ttl=64 time=0.345 ms
64 bytes from 192.168.56.109: icmp_seq=10 ttl=64 time=0.299 ms
64 bytes from 192.168.56.109: icmp_seq=11 ttl=64 time=0.297 ms
64 bytes from 192.168.56.109: icmp_seq=12 ttl=64 time=0.242 ms
64 bytes from 192.168.56.109: icmp_seq=13 ttl=64 time=0.310 ms
64 bytes from 192.168.56.109: icmp_seq=14 ttl=64 time=0.348 ms
64 bytes from 192.168.56.109: icmp_seq=15 ttl=64 time=0.354 ms
64 bytes from 192.168.56.109: icmp_seq=16 ttl=64 time=0.323 ms
64 bytes from 192.168.56.109: icmp_seq=17 ttl=64 time=0.200 ms
64 bytes from 192.168.56.109: icmp_seq=18 ttl=64 time=0.541 ms
64 bytes from 192.168.56.109: icmp_seq=19 ttl=64 time=0.255 ms
64 bytes from 192.168.56.109: icmp_seq=20 ttl=64 time=0.341 ms
64 bytes from 192.168.56.109: icmp_seq=21 ttl=64 time=0.363 ms
64 bytes from 192.168.56.109: icmp_seq=22 ttl=64 time=0.279 ms
```

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

```
[2]+  Stopped                 ping 192.168.56.105
berja@localmachine:~$ ping 192.168.56.110
PING 192.168.56.110 (192.168.56.110) 56(84) bytes of data.
64 bytes from 192.168.56.110: icmp_seq=1 ttl=64 time=0.905 ms
64 bytes from 192.168.56.110: icmp_seq=2 ttl=64 time=0.273 ms
64 bytes from 192.168.56.110: icmp_seq=3 ttl=64 time=0.209 ms
64 bytes from 192.168.56.110: icmp_seq=4 ttl=64 time=0.244 ms
64 bytes from 192.168.56.110: icmp_seq=5 ttl=64 time=0.386 ms
```

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

```
berja@bserver1:~$ ping 192.168.56.110
PING 192.168.56.110 (192.168.56.110) 56(84) bytes of data.
64 bytes from 192.168.56.110: icmp_seq=1 ttl=64 time=0.485 ms
64 bytes from 192.168.56.110: icmp_seq=2 ttl=64 time=0.345 ms
64 bytes from 192.168.56.110: icmp_seq=3 ttl=64 time=0.229 ms
64 bytes from 192.168.56.110: icmp_seq=4 ttl=64 time=0.273 ms
64 bytes from 192.168.56.110: icmp_seq=5 ttl=64 time=0.238 ms
64 bytes from 192.168.56.110: icmp_seq=6 ttl=64 time=0.333 ms
64 bytes from 192.168.56.110: icmp_seq=7 ttl=64 time=0.254 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

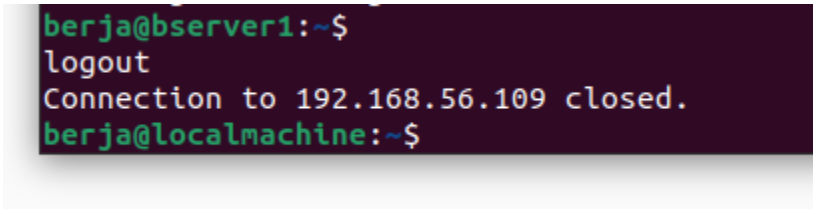
```
berja@localmachine:~$ ssh berja@192.168.56.109
The authenticity of host '192.168.56.109 (192.168.56.109)' can't be established.
ED25519 key fingerprint is SHA256:YU2zYAmfEhMKhhnQBYl6NmJlEnTlVUnFvpo58eZO8Ps.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.109' (ED25519) to the list of known hosts
.
berja@192.168.56.109's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

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

  System information as of Wed Aug 23 15:48:21 UTC 2023

  System load:  0.0               Processes:               105
  Usage of /:   36.4% of 8.02GB   Users logged in:         1
  Memory usage: 5%                IPv4 address for enp0s3: 192.168.56.109
  Swap usage:   0%
```

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

  System information as of Wed Aug 23 15:48:21 UTC 2023

  System load:  0.0               Processes:               105
  Usage of /:   36.4% of 8.02GB   Users logged in:         1
  Memory usage: 5%                IPv4 address for enp0s3: 192.168.56.109
  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

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


Last login: Wed Aug 23 15:40:15 2023
```

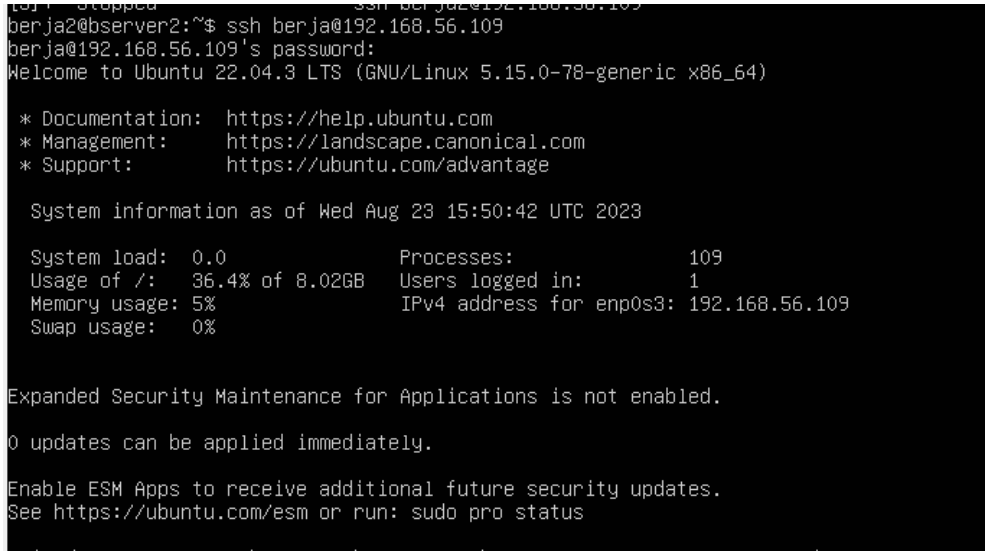
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

```
berja@bserver1:~$ _
```

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



3. Do the same for Server 2.



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.

```
GNU nano 6.2                                    /etc
127.0.0.1       localhost
127.0.0.1       berja-VirtualBox
192.168.56.109  berja
192.168.56.110  berja2

# The following lines are desirable for I
::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.

```
berja@localmachine:~$ ssh berja@berja
The authenticity of host 'berja (192.168.56.109)' can't be established.
ED25519 key fingerprint is SHA256:YU2zYAmfEhMKhhnQBYl6NmJlEnTlVUnFvpo58eZO8Ps.
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])?
```

```
Host key verification failed.
berja@localmachine:~$ ssh berja@berja2
The authenticity of host 'berja2 (192.168.56.110)' can't be established.
ED25519 key fingerprint is SHA256:s9k8cvOuqQfW1ezfZ05kJnN02oGJT4/iY4PY5WMcrSk
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   Because we added their individual IP addresses to the hosts that the local machine has access to, we were able to utilise the hostname to access the control node using SSH commands. Every time we called their hostname as a result, the discovered IP address was utilised.

2. How secured is SSH?

   Because only the local computer and anybody with access to the control nodes' IP addresses may view the data on each command and instruction, SSH is incredibly secure. Because some people can access your computer files via the IP address itself, protecting your IP address is crucial.

conclusion:

I was able to learn about the SSH server and how to use the SSH command to link three distinct systems together through their CLI in this lab exercise. This enables us to quickly access distant units that would otherwise require extensive time and effort to travel to. This shortens the process and improves productivity. This is crucial for companies that link numerous machines to a single server. With only one click, we can more quickly and easily deploy programmes and scripts to other computers.