Name: Bello, Ian Carlo T.	**Date Performed:** August 24, 2022
Course/Section: CPE 232-CPE31S24	**Date Submitted:** August 24, 2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem – 3rd year

Activity 1: Configure Network using Virtual Machines

1. Objectives:

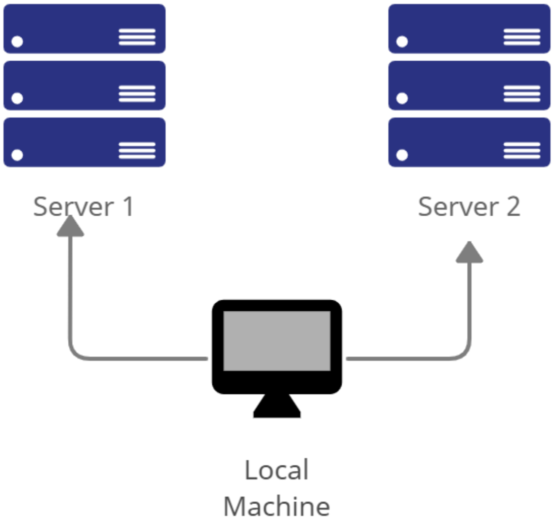
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. Discussion:

Network Topology:

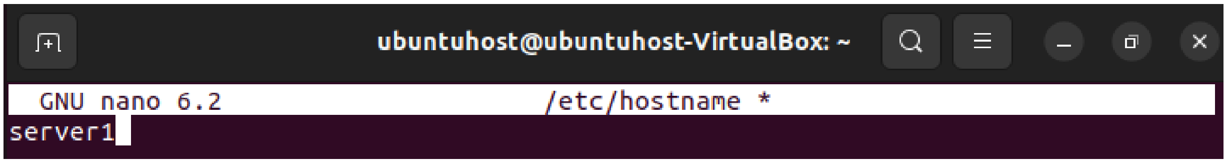
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



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.

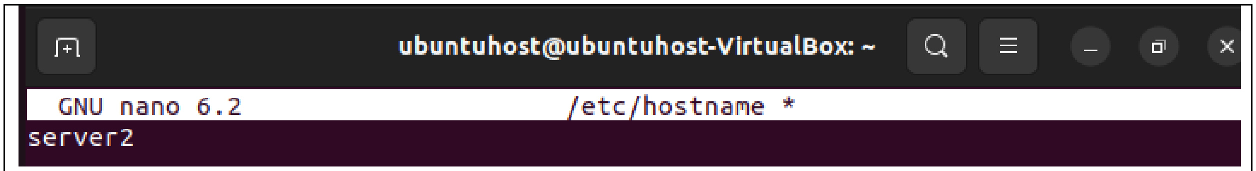
1. Change the hostname using the command *sudo nano /etc/hostname*

   1.1 Use server1 for Server 1



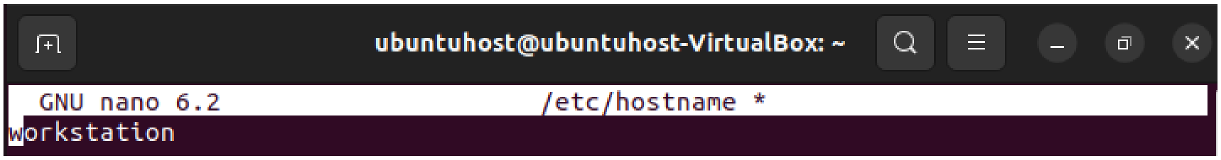
```
  GNU nano 6.2                     /etc/hostname *
server1
```

   1.2 Use server2 for Server 2



```
  GNU nano 6.2                    /etc/hostname *
server2
```

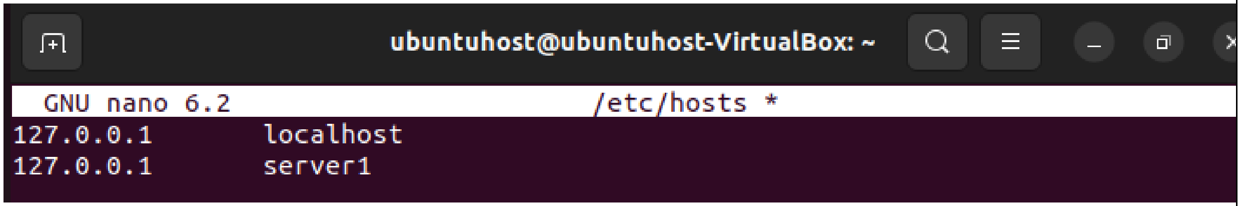
1.3 Use workstation for the Local Machine



```
  GNU nano 6.2                    /etc/hostname *
workstation
```

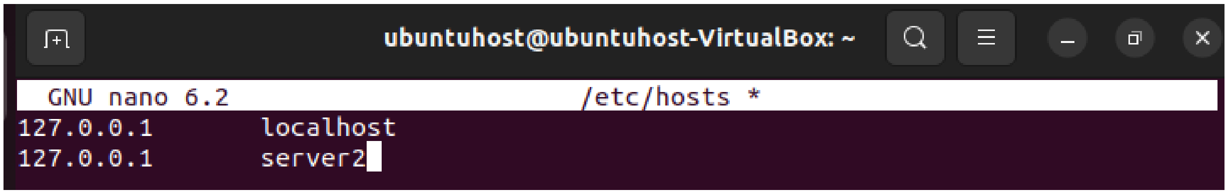
2. Edit the hosts using the command *sudo nano /etc/hosts.* Edit the second line.

2.1 Type 127.0.0.1 server 1 for Server 1



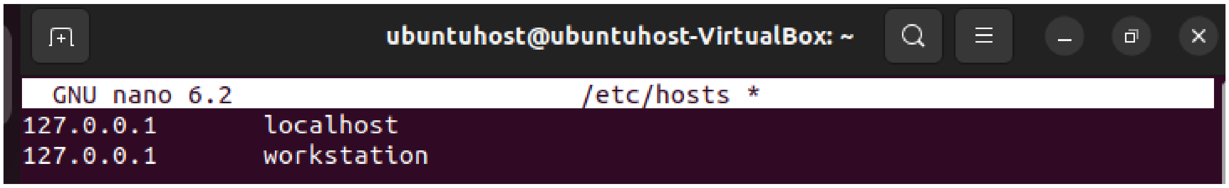
```
  GNU nano 6.2                    /etc/hosts *
127.0.0.1       localhost
127.0.0.1       server1
```

2.2 Type 127.0.0.1 server 2 for Server 2



```
  GNU nano 6.2                    /etc/hosts *
127.0.0.1       localhost
127.0.0.1       server2
```

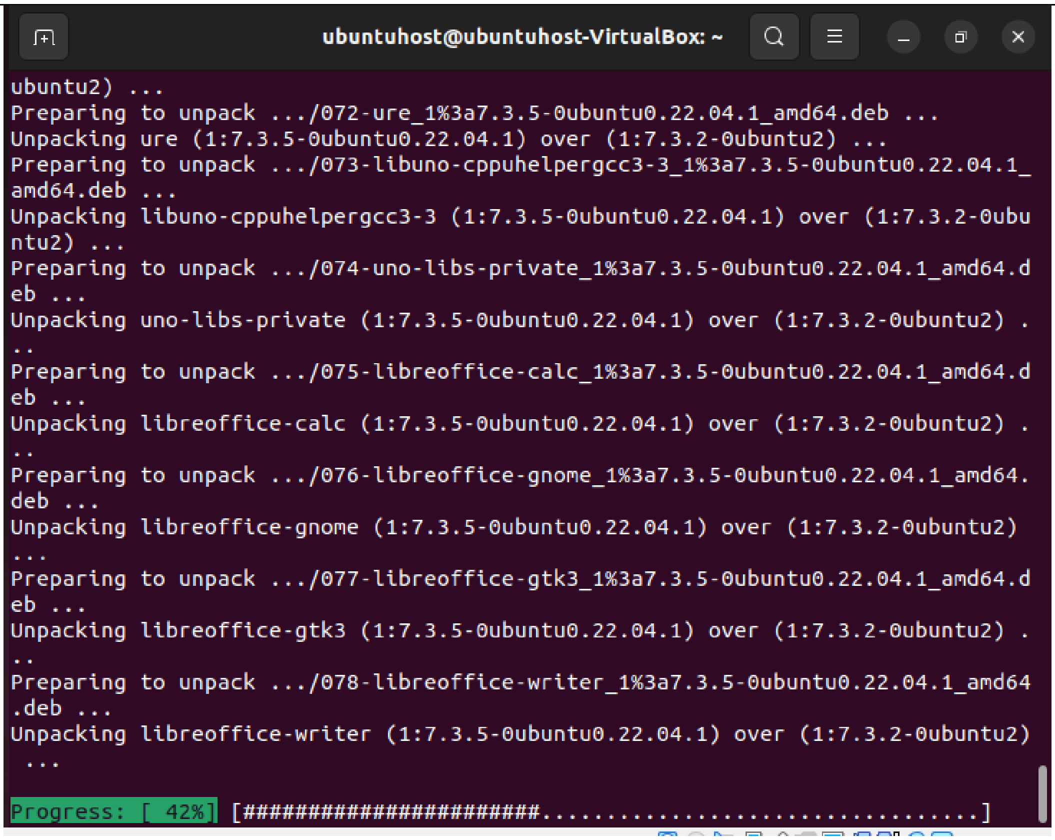
2.3 Type 127.0.0.1 workstation for the Local Machine



```
  GNU nano 6.2                    /etc/hosts *
127.0.0.1       localhost
127.0.0.1       workstation
```

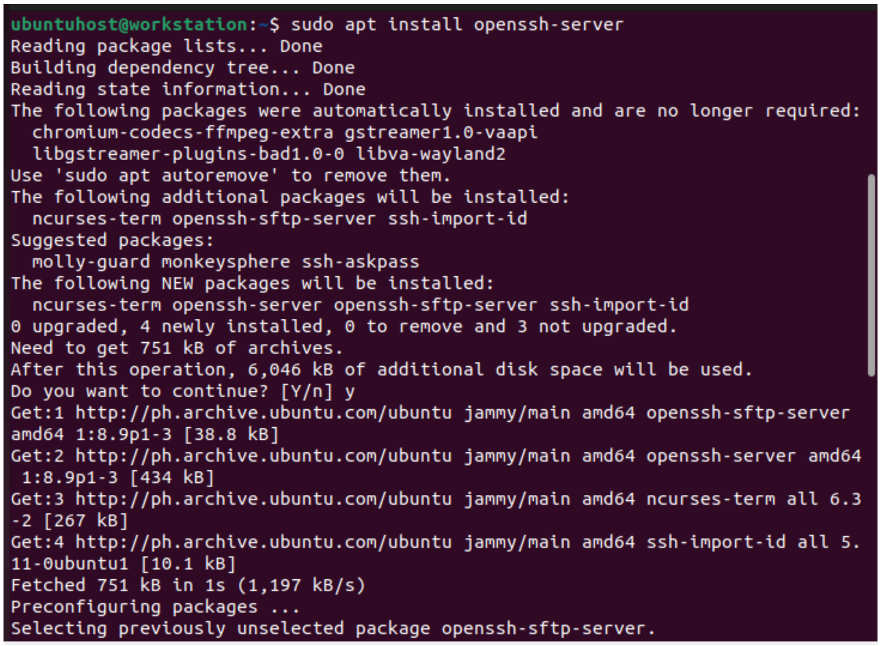
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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



```
ubuntu2) ...
Preparing to unpack .../072-ure_1%3a7.3.5-0ubuntu0.22.04.1_amd64.deb ...
Unpacking ure (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2) ...
Preparing to unpack .../073-libuno-cppuhelpergcc3-3_1%3a7.3.5-0ubuntu0.22.04.1_
amd64.deb ...
Unpacking libuno-cppuhelpergcc3-3 (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubu
ntu2) ...
Preparing to unpack .../074-uno-libs-private_1%3a7.3.5-0ubuntu0.22.04.1_amd64.d
eb ...
Unpacking uno-libs-private (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2) .
..
Preparing to unpack .../075-libreoffice-calc_1%3a7.3.5-0ubuntu0.22.04.1_amd64.d
eb ...
Unpacking libreoffice-calc (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2) .
..
Preparing to unpack .../076-libreoffice-gnome_1%3a7.3.5-0ubuntu0.22.04.1_amd64.
deb ...
Unpacking libreoffice-gnome (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2)
...
Preparing to unpack .../077-libreoffice-gtk3_1%3a7.3.5-0ubuntu0.22.04.1_amd64.d
eb ...
Unpacking libreoffice-gtk3 (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2) .
..
Preparing to unpack .../078-libreoffice-writer_1%3a7.3.5-0ubuntu0.22.04.1_amd64
.deb ...
Unpacking libreoffice-writer (1:7.3.5-0ubuntu0.22.04.1) over (1:7.3.2-0ubuntu2)
 ...

Progress: [ 42%] [#######################................................] 
```

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

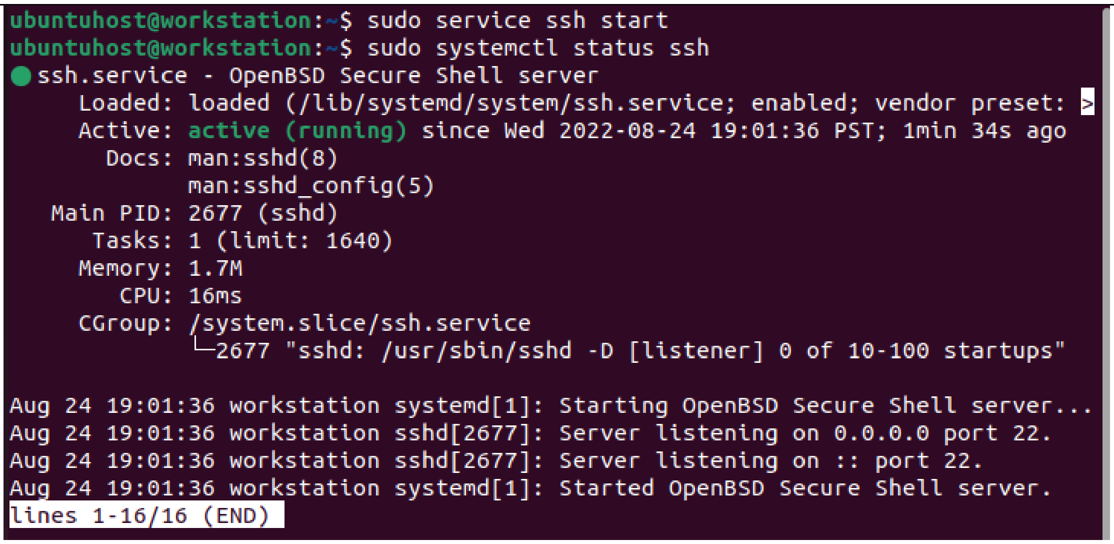


```
ubuntuhost@workstation:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
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/main amd64 openssh-sftp-server
amd64 1:8.9p1-3 [38.8 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 openssh-server amd64
 1:8.9p1-3 [434 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ncurses-term all 6.3
-2 [267 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.
11-0ubuntu1 [10.1 kB]
Fetched 751 kB in 1s (1,197 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
```

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

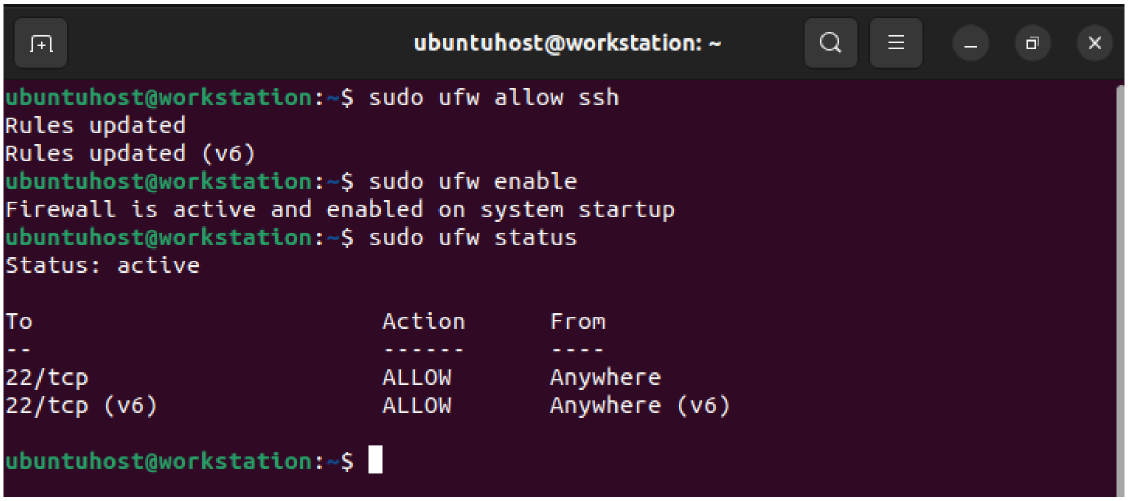
   3.1 *sudo service ssh start*

   3.2 *sudo systemctl status ssh*



```
ubuntuhost@workstation:~$ sudo service ssh start
ubuntuhost@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Wed 2022-08-24 19:01:36 PST; 1min 34s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 2677 (sshd)
      Tasks: 1 (limit: 1640)
     Memory: 1.7M
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─2677 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 24 19:01:36 workstation systemd[1]: Starting OpenBSD Secure Shell server...
Aug 24 19:01:36 workstation sshd[2677]: Server listening on 0.0.0.0 port 22.
Aug 24 19:01:36 workstation sshd[2677]: Server listening on :: port 22.
Aug 24 19:01:36 workstation systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

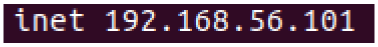
4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*



```
ubuntuhost@workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
ubuntuhost@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntuhost@workstation:~$ sudo ufw status
Status: active

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

ubuntuhost@workstation:~$ 
```

Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

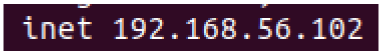
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.
   1.1 Server 1 IP address: 192.168.56.101



```
inet 192.168.56.101
```

this is server1

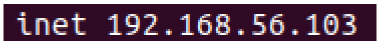
   1.2 Server 2 IP address: 192.168.56.102



```
inet 192.168.56.102
```

this is workstation

   1.3 Server 3 IP address: 192.168.56.103

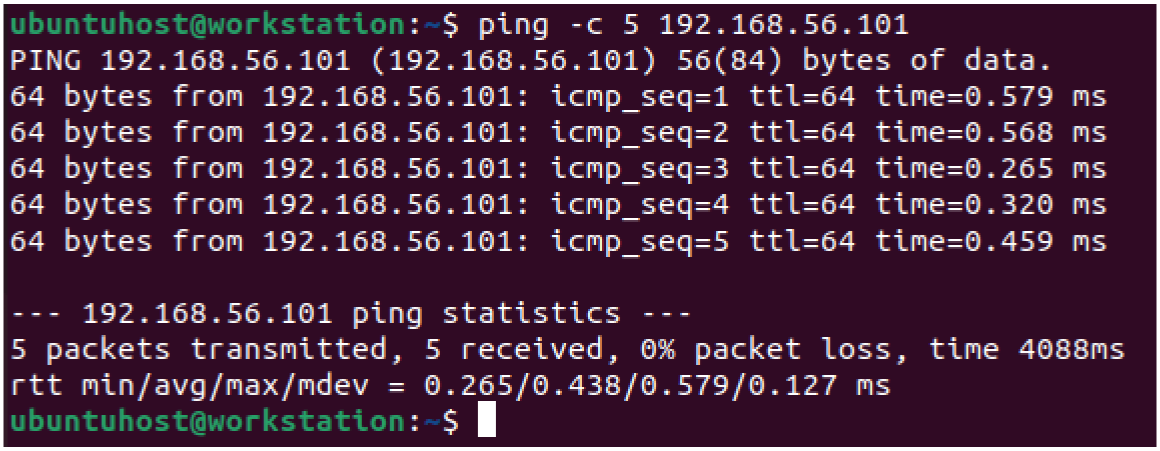


```
inet 192.168.56.103
```

this is server2

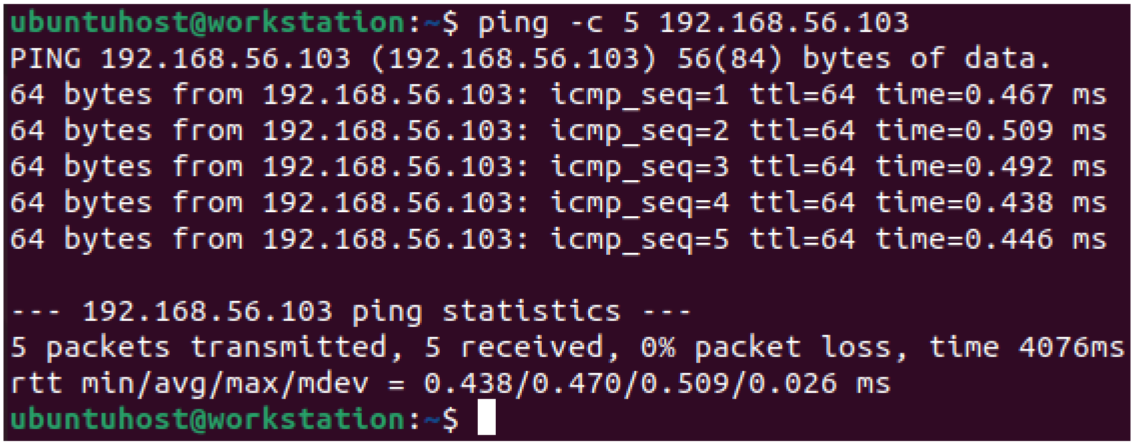
2. Make sure that they can ping each other.

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



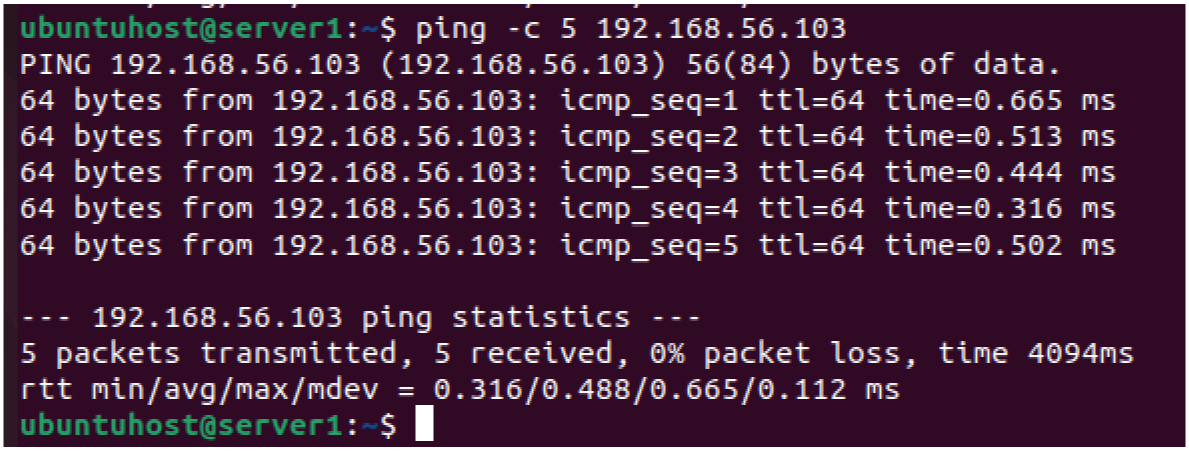
```
ubuntuhost@workstation:~$ ping -c 5 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.579 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.568 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.265 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.320 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.459 ms

--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4088ms
rtt min/avg/max/mdev = 0.265/0.438/0.579/0.127 ms
ubuntuhost@workstation:~$
```

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



```
ubuntuhost@workstation:~$ ping -c 5 192.168.56.103
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=0.467 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.509 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.492 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.438 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.446 ms

--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.438/0.470/0.509/0.026 ms
ubuntuhost@workstation:~$
```

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

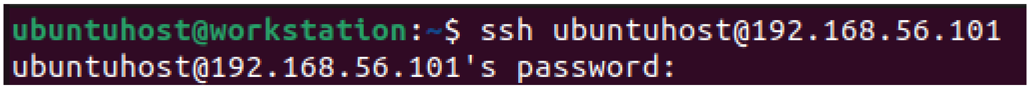


```
ubuntuhost@server1:~$ ping -c 5 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.665 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.513 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.444 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.316 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.502 ms

--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4094ms
rtt min/avg/max/mdev = 0.316/0.488/0.665/0.112 ms
ubuntuhost@server1:~$
```

Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

1. On the Local Machine, issue the following commands:

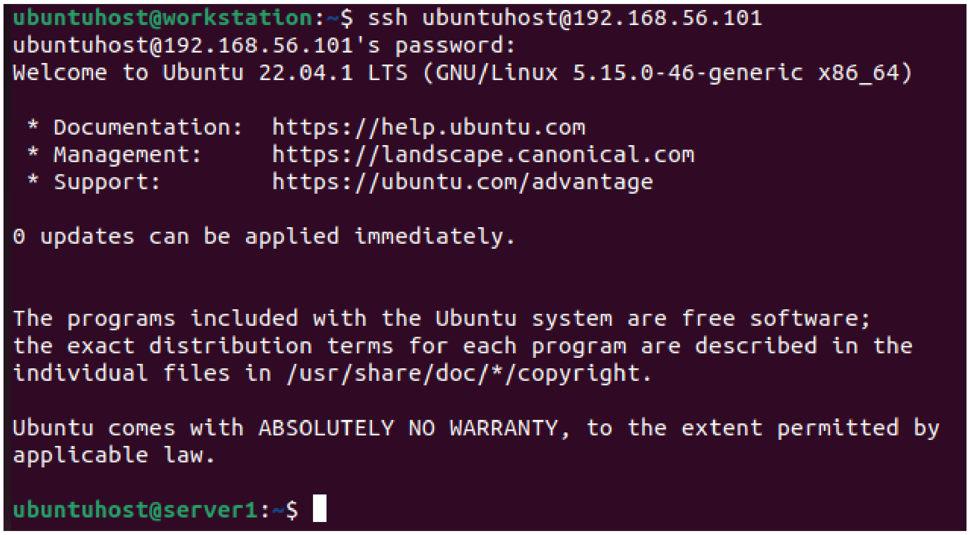
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*



```
ubuntuhost@workstation:~$ ssh ubuntuhost@192.168.56.101
ubuntuhost@192.168.56.101's password:
```

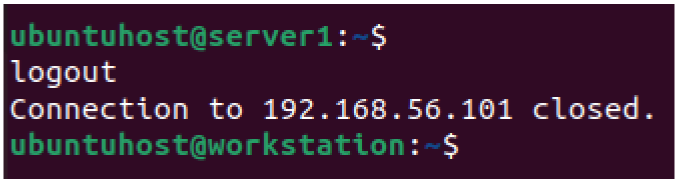
1.2 Enter the password for server 1 when prompted

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



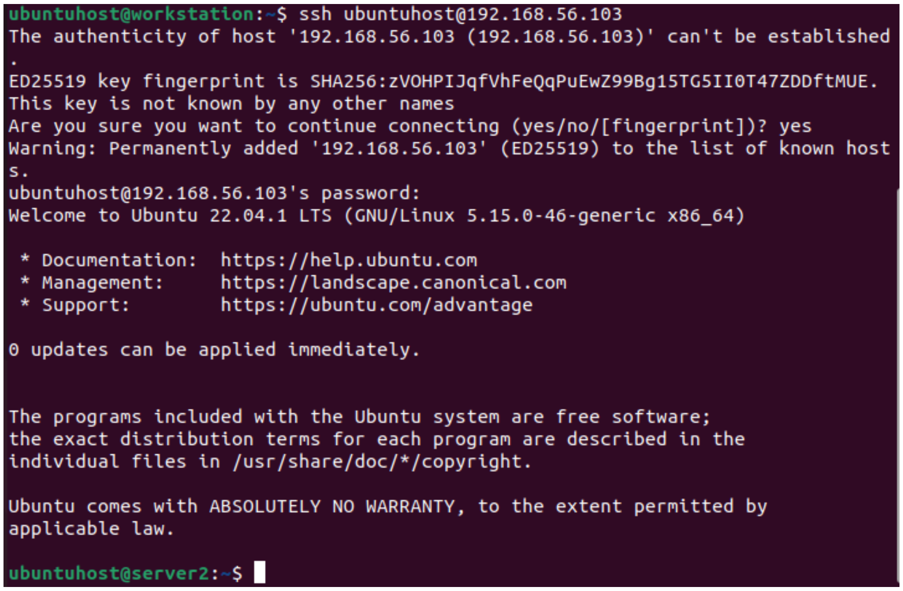
```
ubuntuhost@workstation:~$ ssh ubuntuhost@192.168.56.101
ubuntuhost@192.168.56.101's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * 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.

ubuntuhost@server1:~$ 
```

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



```
ubuntuhost@server1:~$
logout
Connection to 192.168.56.101 closed.
ubuntuhost@workstation:~$
```

3. Do the same for Server 2.

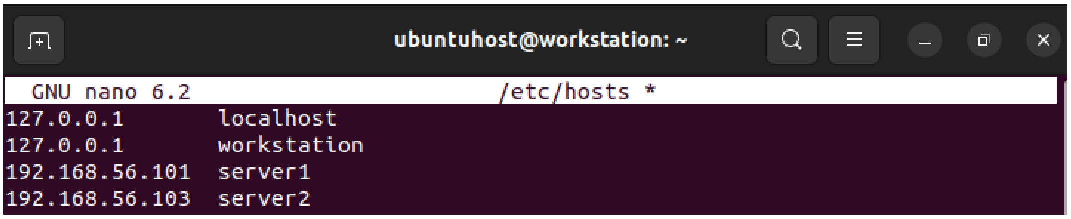


```
ubuntuhost@workstation:~$ ssh ubuntuhost@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established
.
ED25519 key fingerprint is SHA256:zVOHPIJqfVhFeQqPuEwZ99Bg15TG5II0T47ZDDftMUE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known host
s.
ubuntuhost@192.168.56.103's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-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.

ubuntuhost@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)

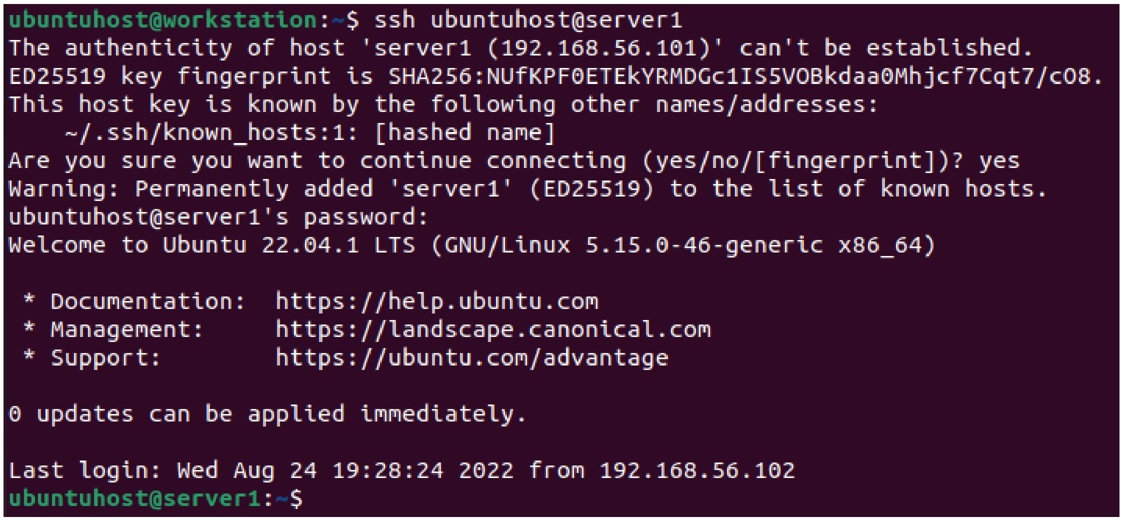
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)



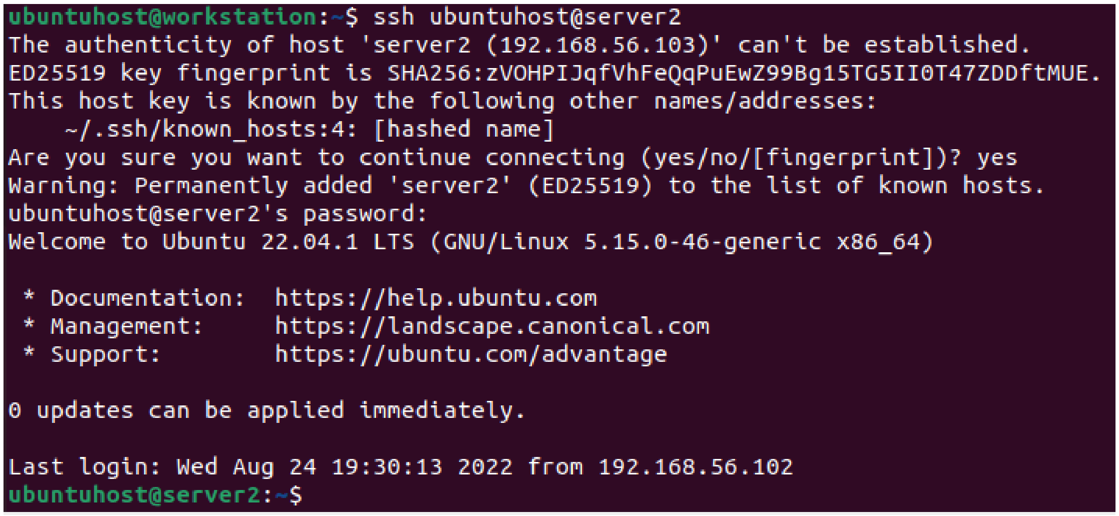
```
 GNU nano 6.2                              /etc/hosts *
127.0.0.1       localhost
127.0.0.1       workstation
192.168.56.101  server1
192.168.56.103  server2
```

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.



```
ubuntuhost@workstation:~$ ssh ubuntuhost@server1
The authenticity of host 'server1 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:NUfKPF0ETEkYRMDGc1IS5VOBkdaa0Mhjcf7Cqt7/c08.
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.
ubuntuhost@server1's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

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

0 updates can be applied immediately.

Last login: Wed Aug 24 19:28:24 2022 from 192.168.56.102
ubuntuhost@server1:~$
```



```
ubuntuhost@workstation:~$ ssh ubuntuhost@server2
The authenticity of host 'server2 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:zVOHPIJqfVhFeQqPuEwZ99Bg15TG5II0T47ZDDftMUE.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:4: [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.
ubuntuhost@server2's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

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

0 updates can be applied immediately.

Last login: Wed Aug 24 19:30:13 2022 from 192.168.56.102
ubuntuhost@server2:~$
```

Reflections:

Answer the following:

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

In the hosts text, we saved the input Ip address from both servers, which will be known information in the workstation where name=value. Ex. Server1=192.168.56.101

2. How secured is SSH?

SSH uses password authentication to work, having remotes to create connection requests securely. SSH encrypts and authenticates all connections. The only problem is the default port, which is port 22. Hackers will try to access this first. Disabling root logins and usage of SSH keys instead of passwords