Name: Rustom C. Cariño	**Date Performed:** 08/22/2023
Course/Section: CPE31S5/CPE 232	**Date Submitted:** 08/23/2023
Instructor: Engr. Roman Richard	**Semester and SY:** 1st sem 2023-2024

Activity 1: Configure Network using Virtual Machines

1. Objectives:

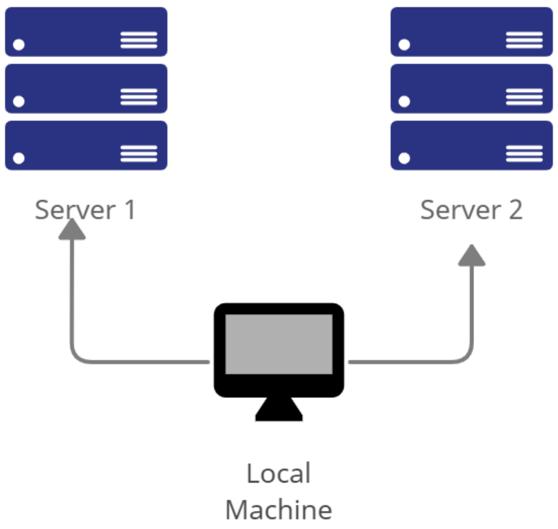
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. Discussion:

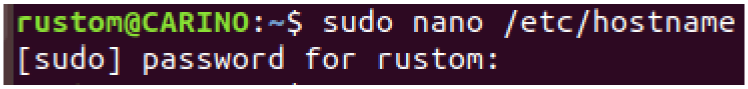
Network Topology:

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*).



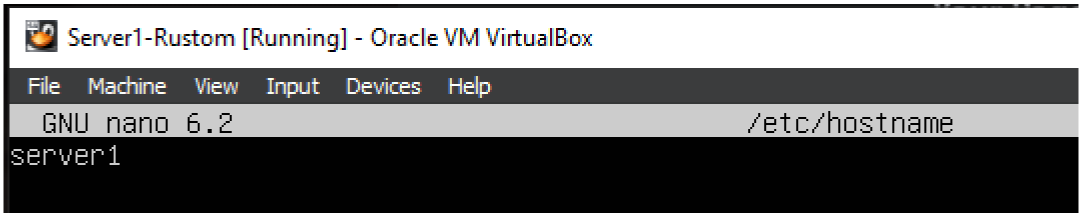
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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



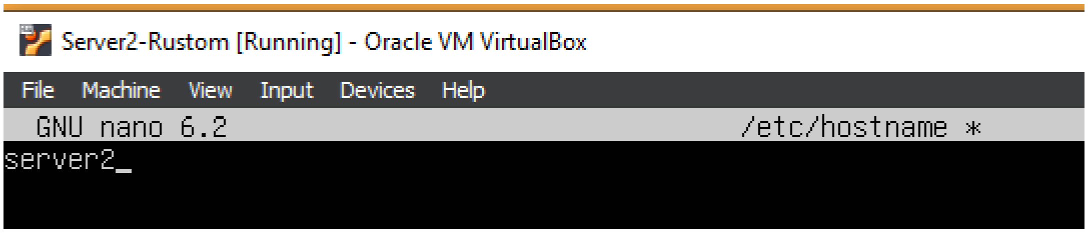
```
rustom@CARINO:~$ sudo nano /etc/hostname
[sudo] password for rustom:
```

1.1Use server1 for Server 1


Server1-Rustom [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

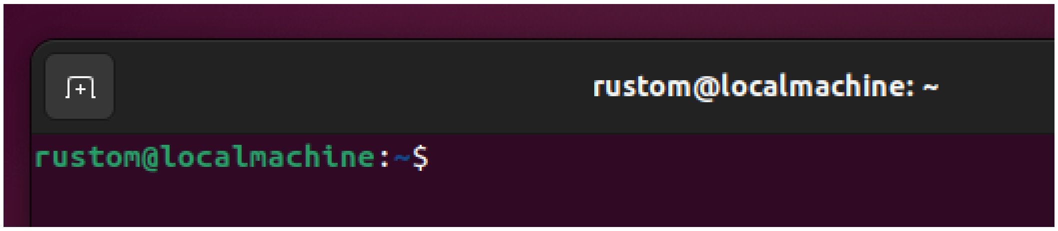
```
  GNU nano 6.2                                        /etc/hostname
server1
```

1.1 Use server2 for Server 2


Server2-Rustom [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
  GNU nano 6.2                                        /etc/hostname *
server2_
```

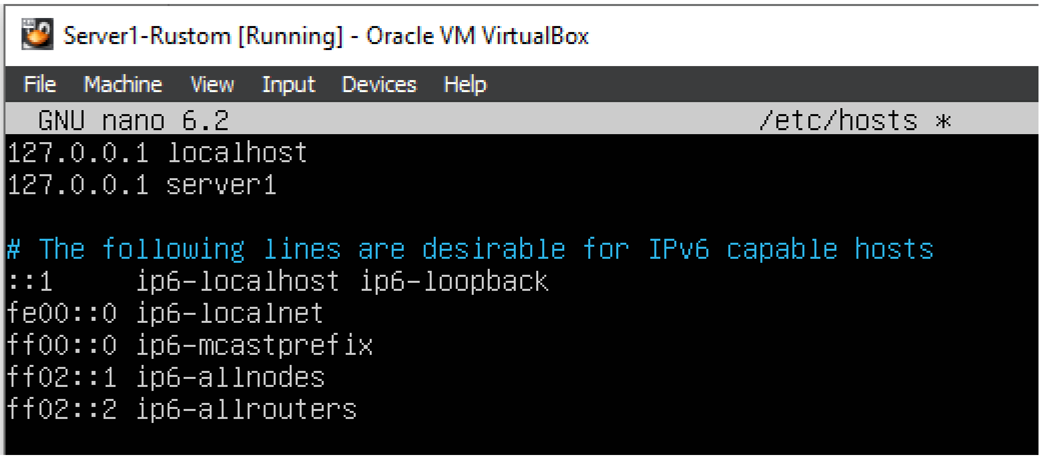
1.2 Use workstation for the Local Machine



```
rustom@localmachine:~$
```

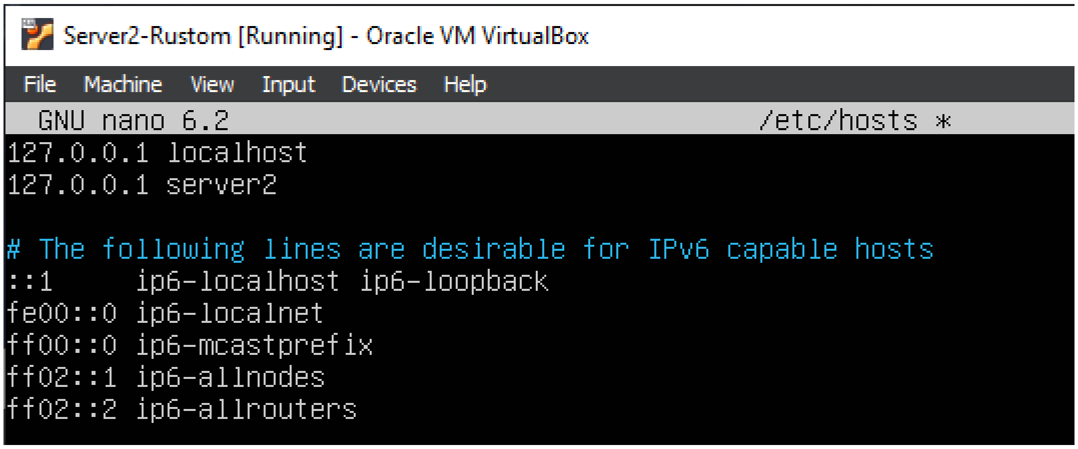
2. Edit the hosts using the command *sudo nano /etc/hosts.* Edit the second line.

2.1 Type 127.0.0.1 server 1 for Server 1


Server1-Rustom [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
  GNU nano 6.2                                          /etc/hosts *
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
```

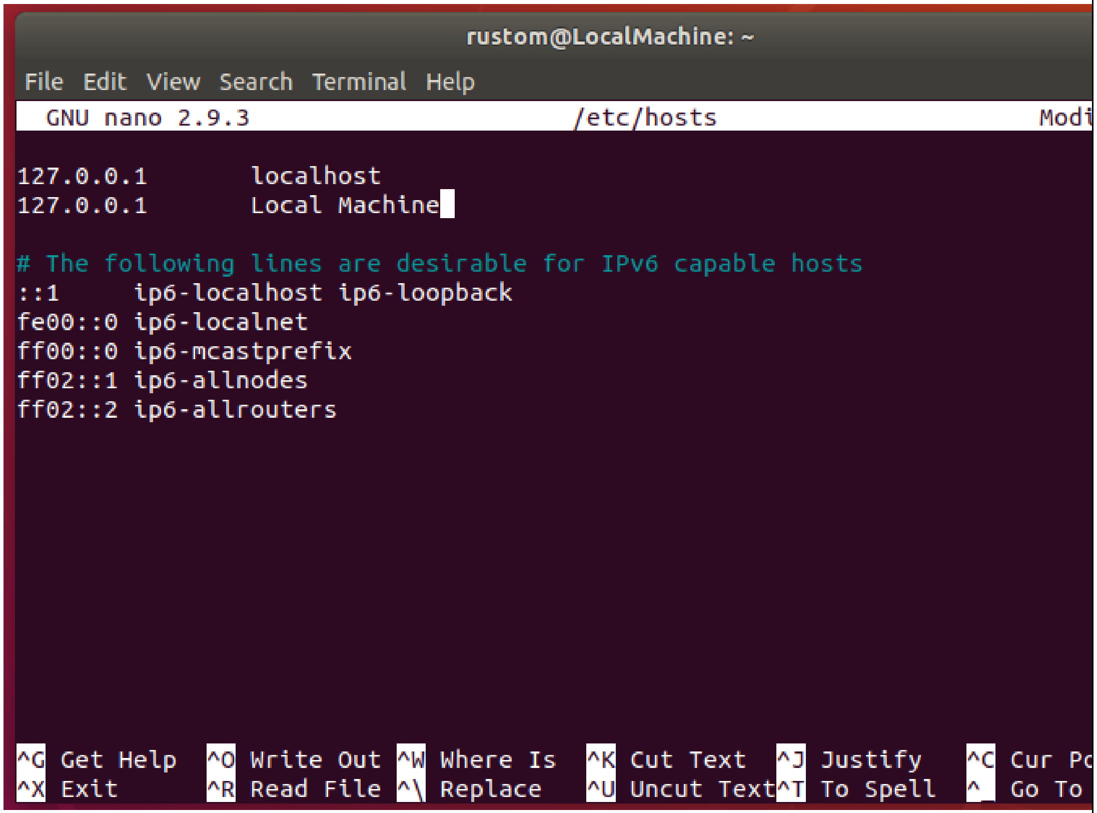
2.2 Type 127.0.0.1 server 2 for Server 2



Server2-Rustom [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
 GNU nano 6.2                                      /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
```

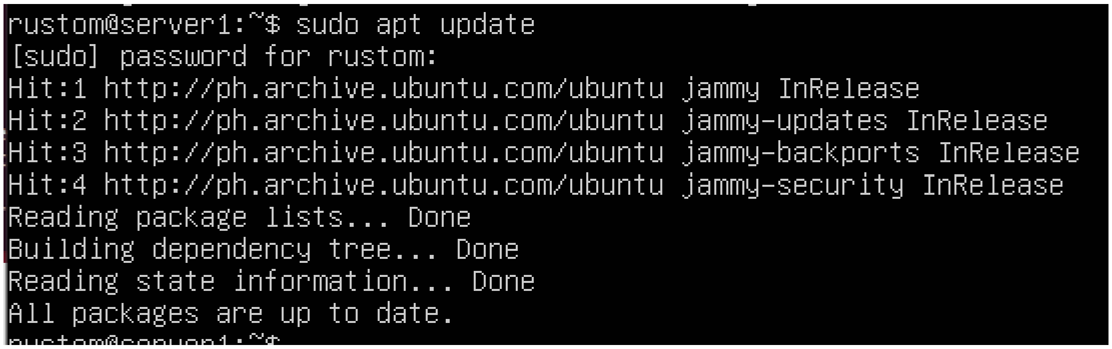
2.3 Type 127.0.0.1 workstation for the Local Machine



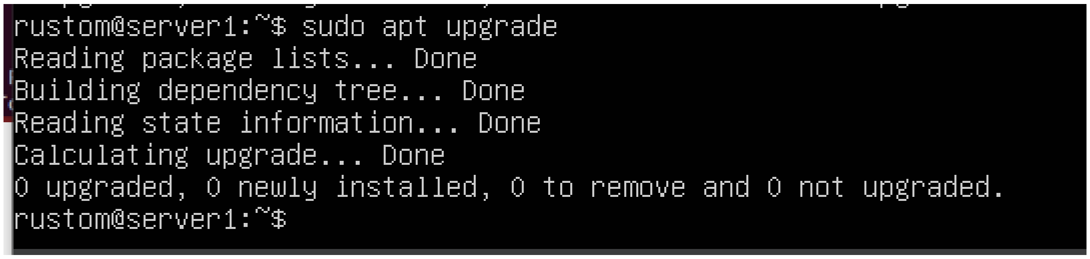
```
  GNU nano 2.9.3                      /etc/hosts                        Modi

127.0.0.1       localhost
127.0.0.1       Local Machine

# 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









^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Po
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To
```

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.

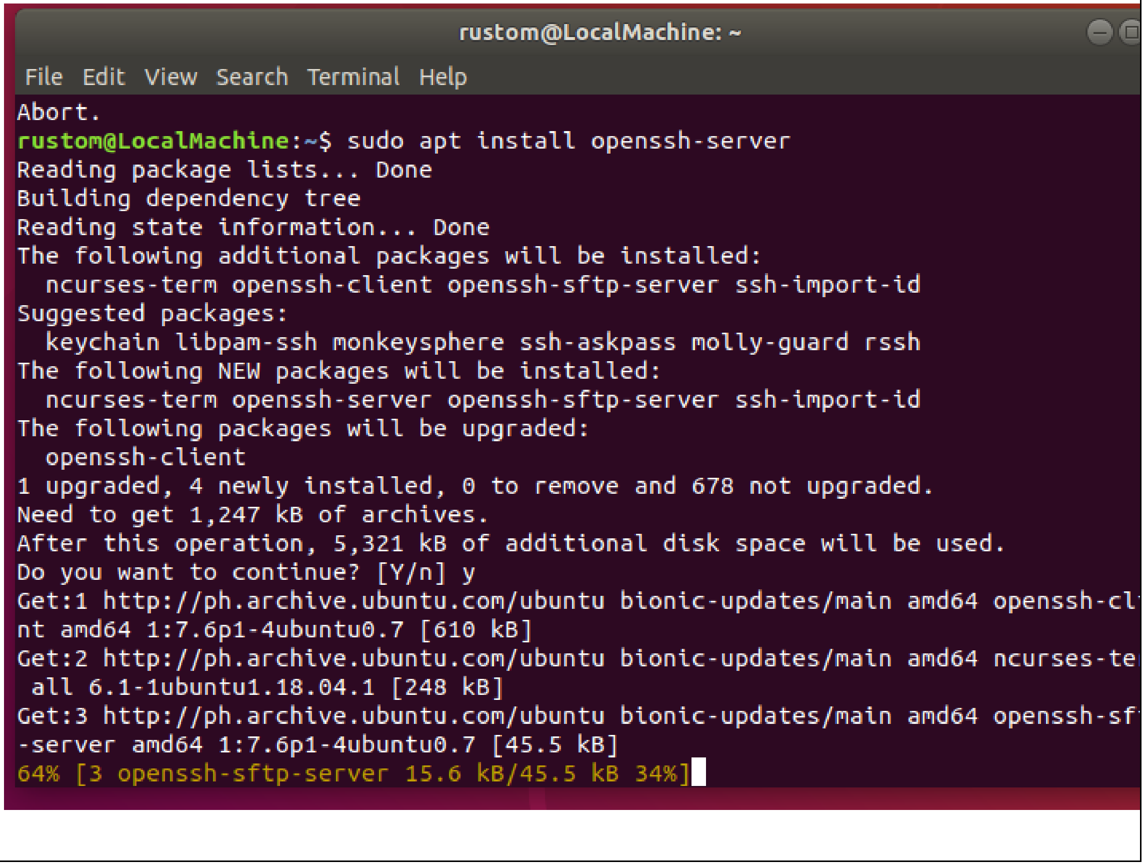


```
rustom@server1:~$ sudo apt update
[sudo] password for rustom:
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
rustom@server1:~$
```



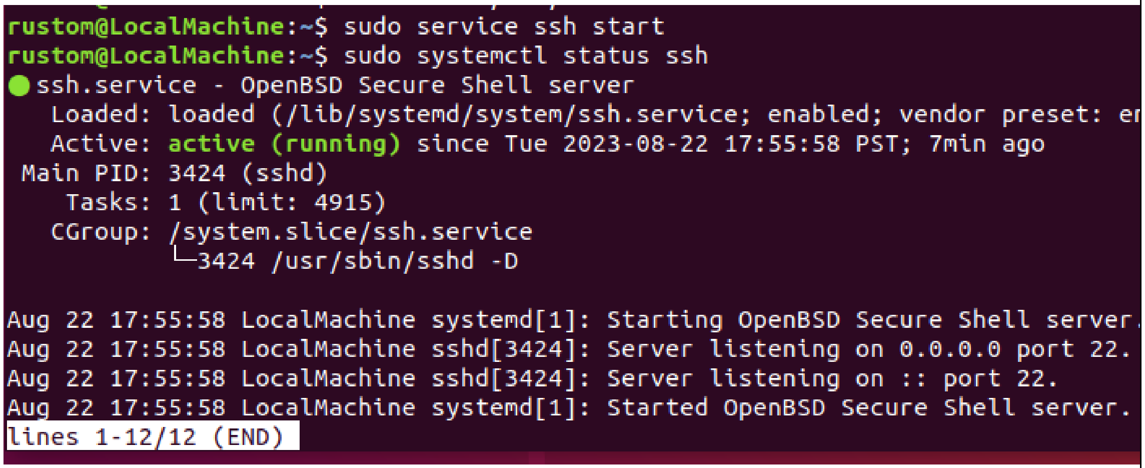
```
rustom@server1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
rustom@server1:~$
```

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

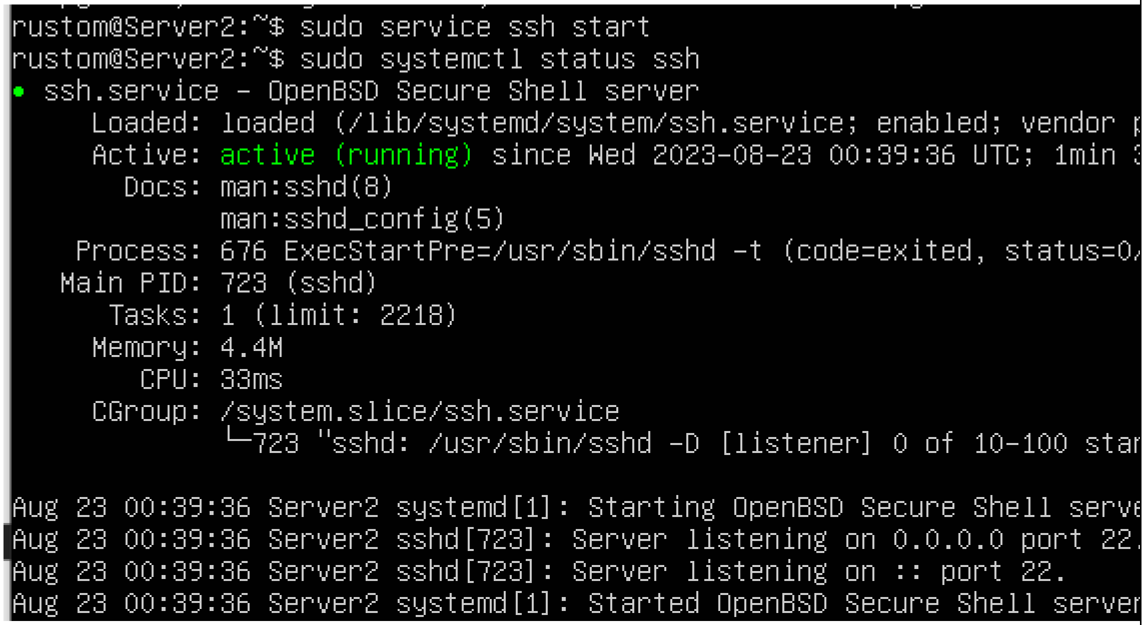


```
Abort.
rustom@LocalMachine:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard rssh
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 678 not upgraded.
Need to get 1,247 kB of archives.
After this operation, 5,321 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openssh-cl
nt amd64 1:7.6p1-4ubuntu0.7 [610 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 ncurses-te
 all 6.1-1ubuntu1.18.04.1 [248 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openssh-sf
-server amd64 1:7.6p1-4ubuntu0.7 [45.5 kB]
64% [3 openssh-sftp-server 15.6 kB/45.5 kB 34%]
```

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*

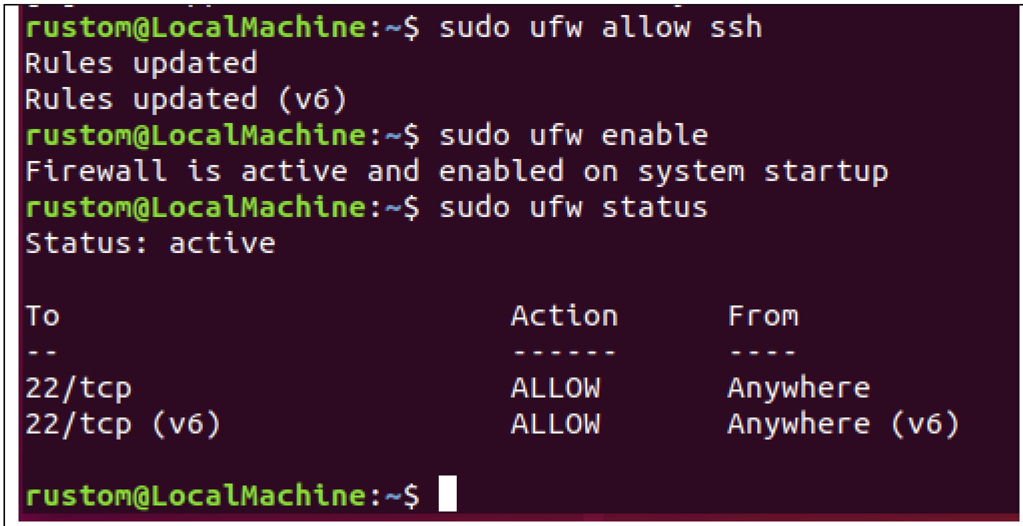


```
rustom@LocalMachine:~$ sudo service ssh start
rustom@LocalMachine:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Tue 2023-08-22 17:55:58 PST; 7min ago
 Main PID: 3424 (sshd)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─3424 /usr/sbin/sshd -D

Aug 22 17:55:58 LocalMachine systemd[1]: Starting OpenBSD Secure Shell server.
Aug 22 17:55:58 LocalMachine sshd[3424]: Server listening on 0.0.0.0 port 22.
Aug 22 17:55:58 LocalMachine sshd[3424]: Server listening on :: port 22.
Aug 22 17:55:58 LocalMachine systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```

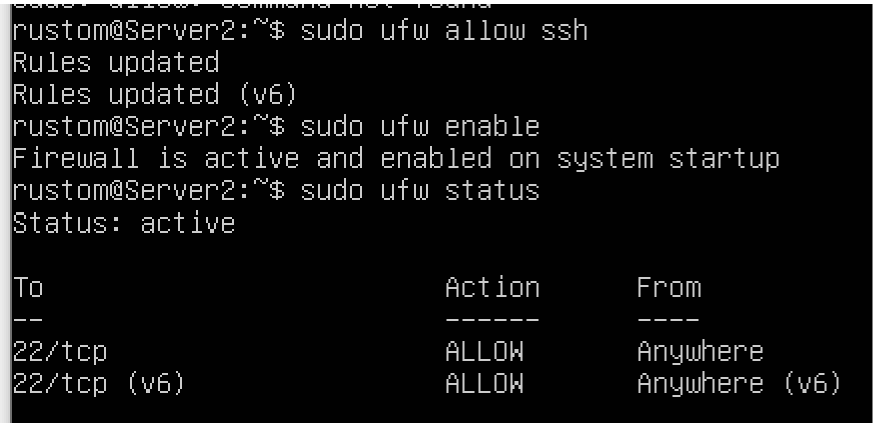


```
rustom@Server2:~$ sudo service ssh start
rustom@Server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor p
     Active: active (running) since Wed 2023-08-23 00:39:36 UTC; 1min 3
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 676 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/
   Main PID: 723 (sshd)
      Tasks: 1 (limit: 2218)
     Memory: 4.4M
        CPU: 33ms
     CGroup: /system.slice/ssh.service
             └─723 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 star

Aug 23 00:39:36 Server2 systemd[1]: Starting OpenBSD Secure Shell serve
Aug 23 00:39:36 Server2 sshd[723]: Server listening on 0.0.0.0 port 22.
Aug 23 00:39:36 Server2 sshd[723]: Server listening on :: port 22.
Aug 23 00:39:36 Server2 systemd[1]: Started OpenBSD Secure Shell server
```

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*



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

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

rustom@LocalMachine:~$
```



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

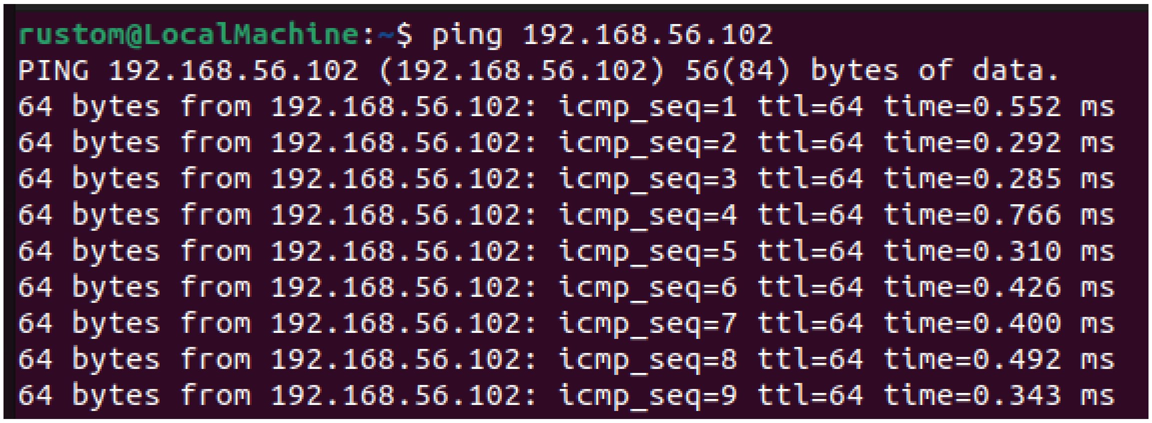
   1.1 Server 1 IP address: 192.168.56.**103**

   1.2 Server 2 IP address: 192.168.56.**102**

   1.3 Server 3 IP address: 192.168.56.**104**

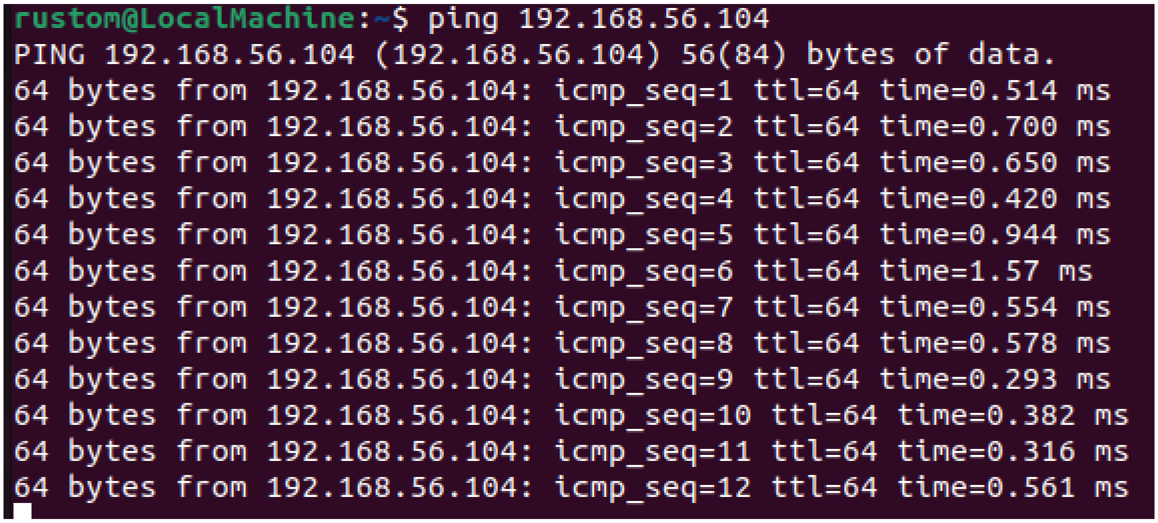
2. Make sure that they can ping each other.

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



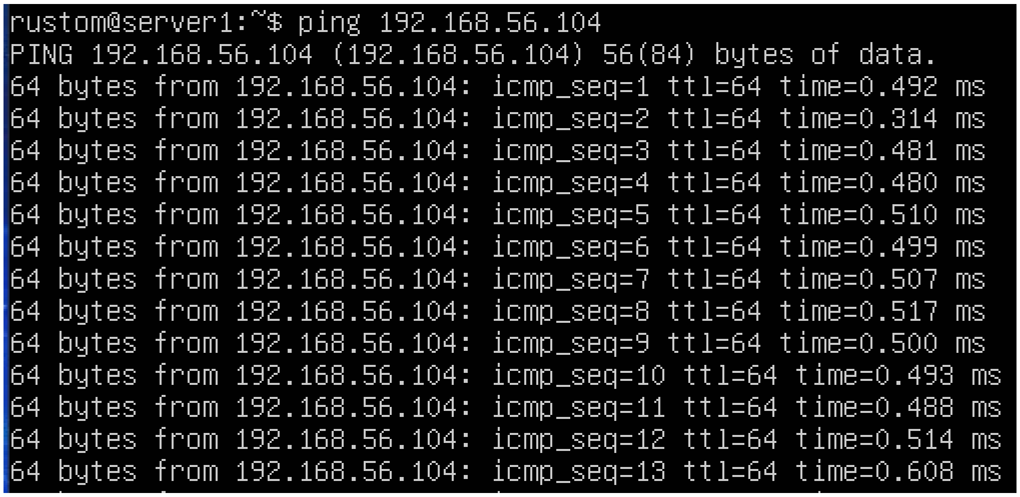
```
rustom@LocalMachine:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.552 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.292 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.285 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.766 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.310 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.426 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.400 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.492 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.343 ms
```

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



```
rustom@LocalMachine:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=0.514 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.650 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.420 ms
64 bytes from 192.168.56.104: icmp_seq=5 ttl=64 time=0.944 ms
64 bytes from 192.168.56.104: icmp_seq=6 ttl=64 time=1.57 ms
64 bytes from 192.168.56.104: icmp_seq=7 ttl=64 time=0.554 ms
64 bytes from 192.168.56.104: icmp_seq=8 ttl=64 time=0.578 ms
64 bytes from 192.168.56.104: icmp_seq=9 ttl=64 time=0.293 ms
64 bytes from 192.168.56.104: icmp_seq=10 ttl=64 time=0.382 ms
64 bytes from 192.168.56.104: icmp_seq=11 ttl=64 time=0.316 ms
64 bytes from 192.168.56.104: icmp_seq=12 ttl=64 time=0.561 ms
```

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

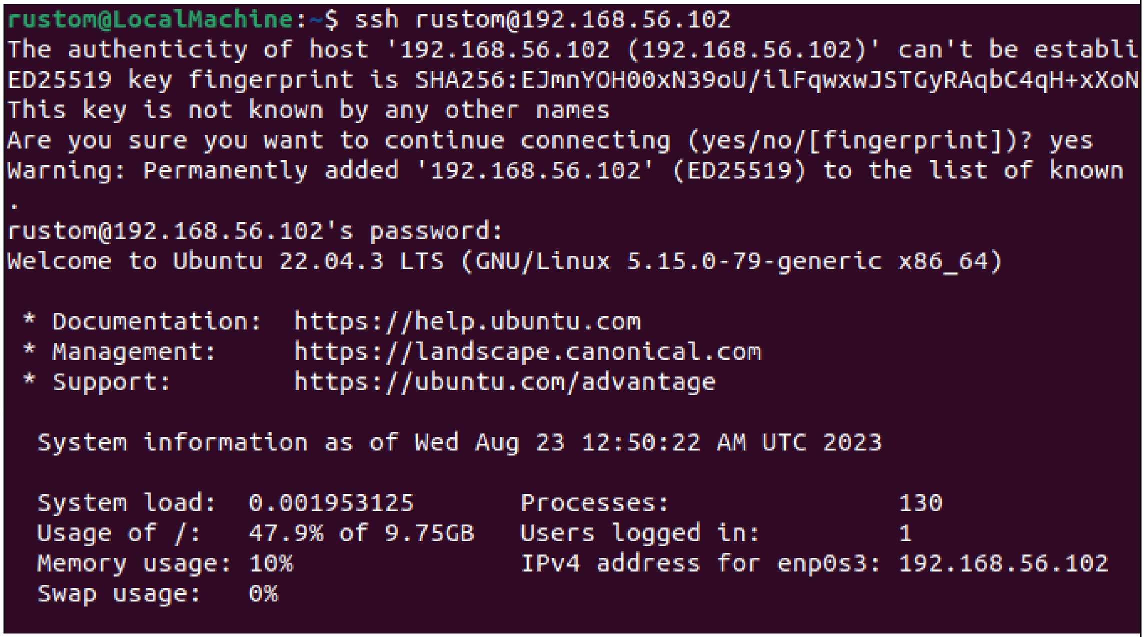


```
rustom@server1:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=0.492 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.314 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.481 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.480 ms
64 bytes from 192.168.56.104: icmp_seq=5 ttl=64 time=0.510 ms
64 bytes from 192.168.56.104: icmp_seq=6 ttl=64 time=0.499 ms
64 bytes from 192.168.56.104: icmp_seq=7 ttl=64 time=0.507 ms
64 bytes from 192.168.56.104: icmp_seq=8 ttl=64 time=0.517 ms
64 bytes from 192.168.56.104: icmp_seq=9 ttl=64 time=0.500 ms
64 bytes from 192.168.56.104: icmp_seq=10 ttl=64 time=0.493 ms
64 bytes from 192.168.56.104: icmp_seq=11 ttl=64 time=0.488 ms
64 bytes from 192.168.56.104: icmp_seq=12 ttl=64 time=0.514 ms
64 bytes from 192.168.56.104: icmp_seq=13 ttl=64 time=0.608 ms
```

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

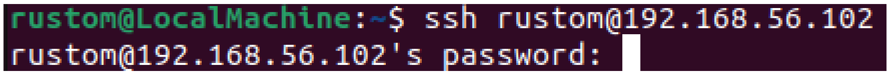
1. On the Local Machine, issue the following commands:

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



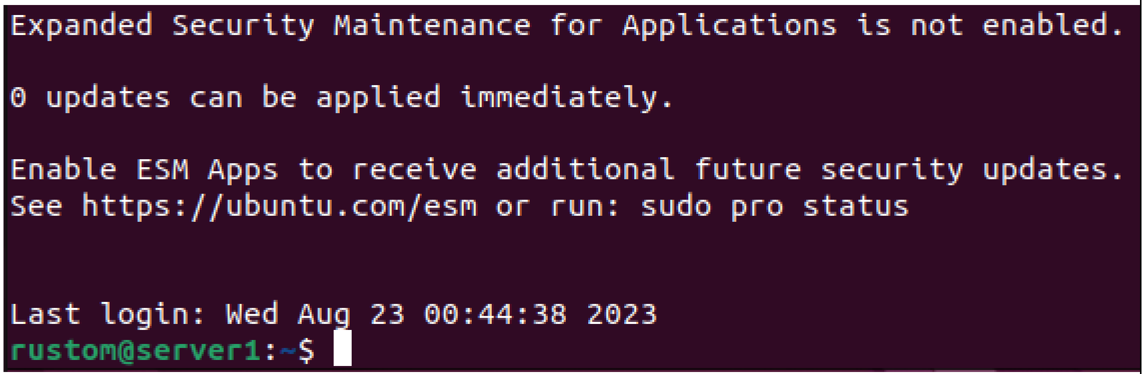
```
rustom@LocalMachine:~$ ssh rustom@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be establi
ED25519 key fingerprint is SHA256:EJmnYOH00xN39oU/ilFqwxwJSTGyRAqbC4qH+xXoN
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known
.
rustom@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 Wed Aug 23 12:50:22 AM UTC 2023

  System load:  0.001953125       Processes:               130
  Usage of /:   47.9% of 9.75GB   Users logged in:         1
  Memory usage: 10%               IPv4 address for enp0s3: 192.168.56.102
  Swap usage:   0%
```

1.2 Enter the password for server 1 when prompted



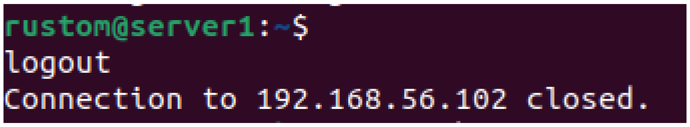
```
rustom@LocalMachine:~$ ssh rustom@192.168.56.102
rustom@192.168.56.102's password:
```

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



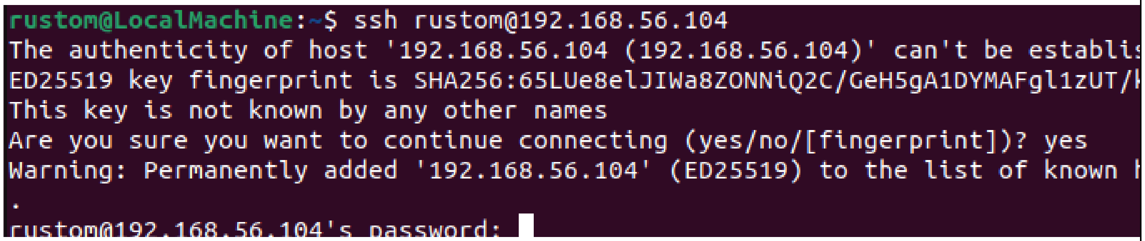
```
Expanded Security Maintenance for Applications is not enabled.

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: Wed Aug 23 00:44:38 2023
rustom@server1:~$ 
```

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



```
rustom@server1:~$
logout
Connection to 192.168.56.102 closed.
```

3. Do the same for Server 2.

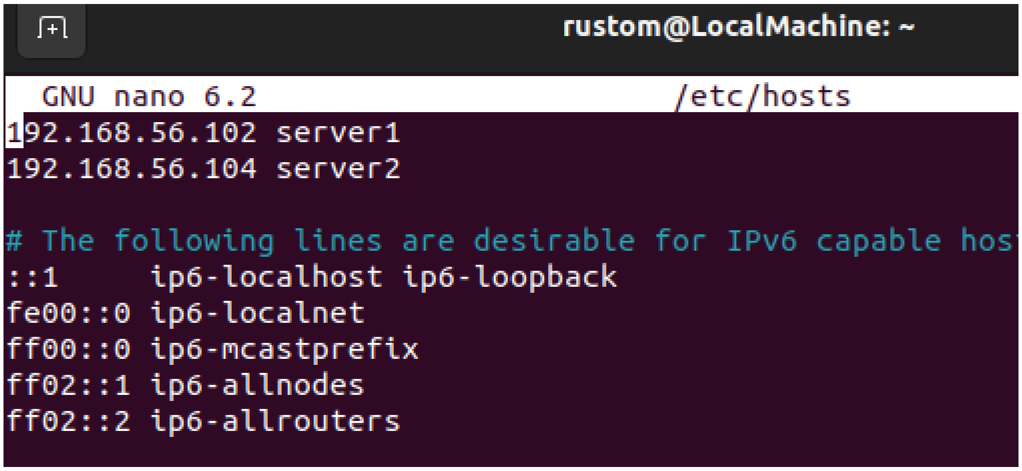


```
rustom@LocalMachine:~$ ssh rustom@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be establis
ED25519 key fingerprint is SHA256:65LUe8elJIWa8ZONNiQ2C/GeH5gA1DYMAFgl1zUT/
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known
.
rustom@192.168.56.104's password: 
```

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)

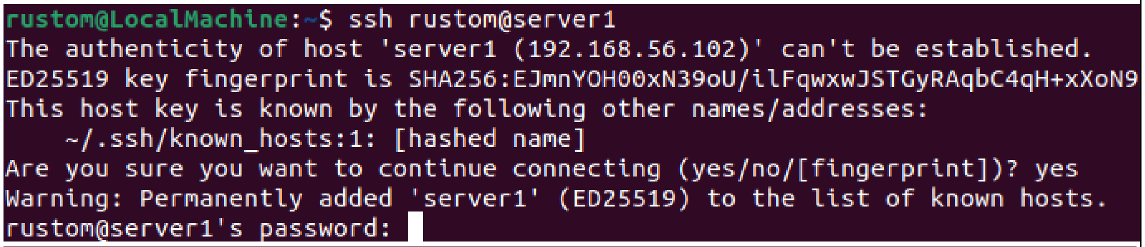
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)



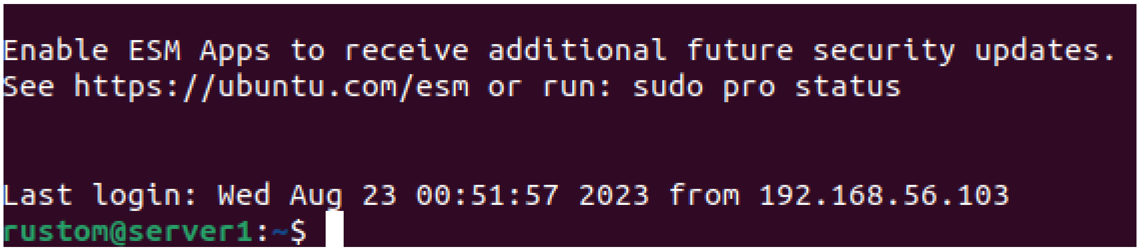
```
  GNU nano 6.2                          /etc/hosts
192.168.56.102 server1
192.168.56.104 server2

# The following lines are desirable for IPv6 capable hos
::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.

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.



```
rustom@LocalMachine:~$ ssh rustom@server1
The authenticity of host 'server1 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:EJmnYOH00xN39oU/ilFqwxwJSTGyRAqbC4qH+xXoN9
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.
rustom@server1's password: 
```



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


Last login: Wed Aug 23 00:51:57 2023 from 192.168.56.103
rustom@server1:~$ 
```

Reflections:

Answer the following:

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

- **We were be able to use the hostname of the server instead of IP address in SSH command because we edited the host of local machine using the command sudo nano /etc/hosts by inputting the ip address of respective server followed by the hostname.**

- How secured is SSH?

- **SSH is absolutely secured since the local machine has a secure access and manage to remote the servers or other machines, allowing the local machine to login to the remote servers and can execute different commands or transfer files as if they were exactly one machine.**

Conclusion:

- **In these hands-on activities I will be able to create and configure virtual machines in virtual box and set up a virtual network as well as testing connectivity of three virtual machines. Using the CLI along with the SSH command I can go through different virtual machine at ease that can be very efficiently when it comes to accessing different units in different location since it wouldn't take a lot of time and effort to travel from one place to another.**