Name: Aducal, John Mark S.	Date Performed: 08 / 21 / 2022
Course/Section: CPE232-CPE31S24	Date Submitted: 08 / 23 / 2022
Instructor: Engr. Jonathan V. Taylar	Semester and SY: 1st Semester SY 2022-2023

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

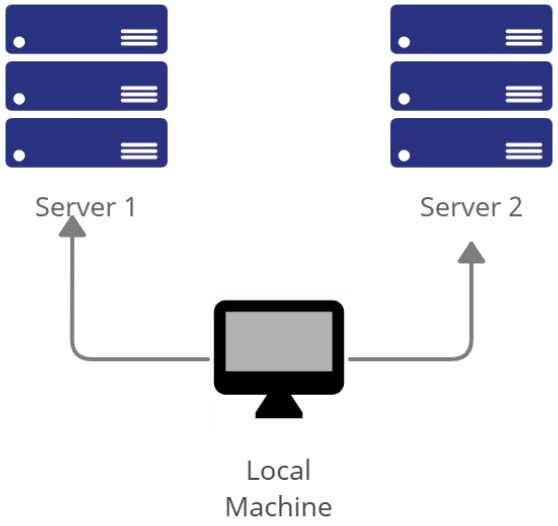
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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*).

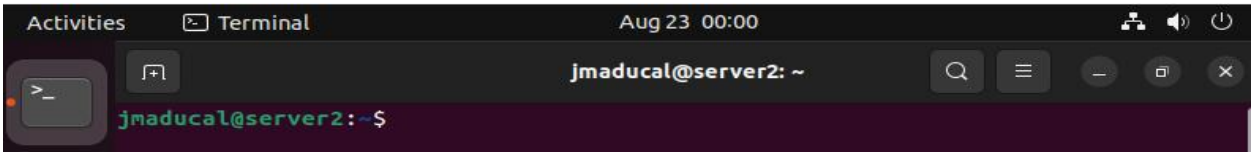


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

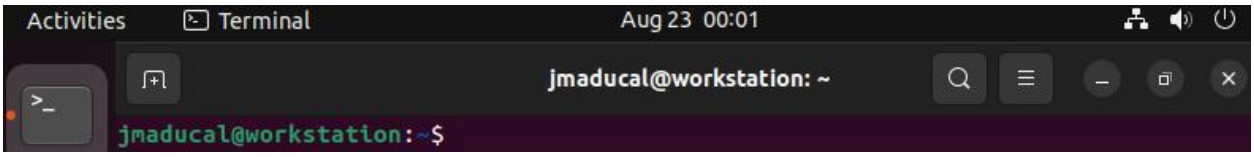
1. Change the hostname using the command *sudo nano /etc/hostname*

1.1 Use server1 for Server 1

1.2 Use server2 for Server 2

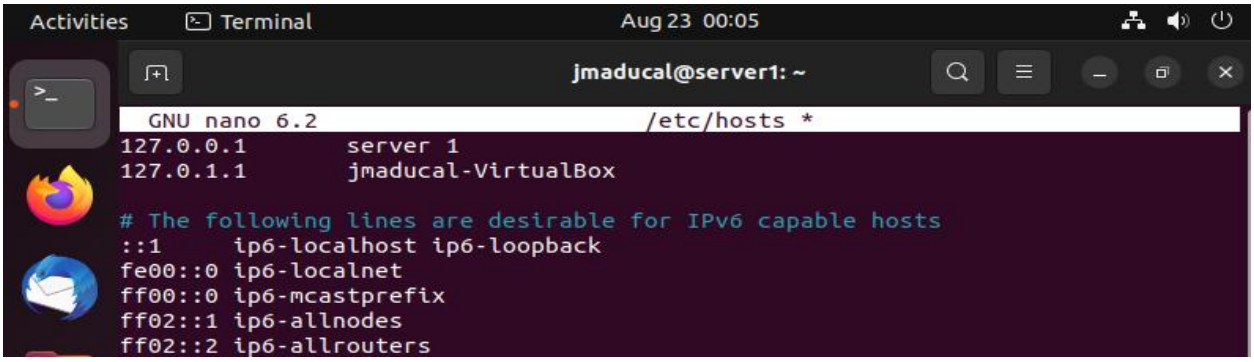


1.3 Use workstation for the Local Machine

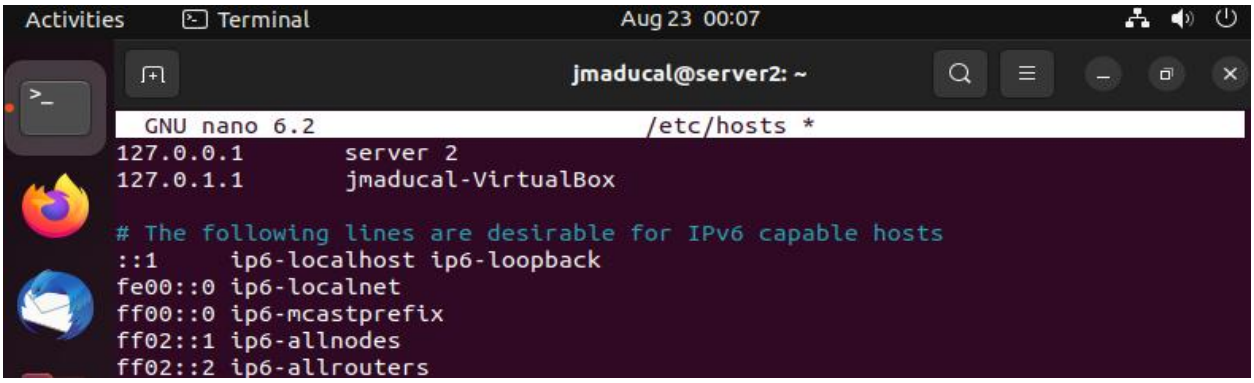


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

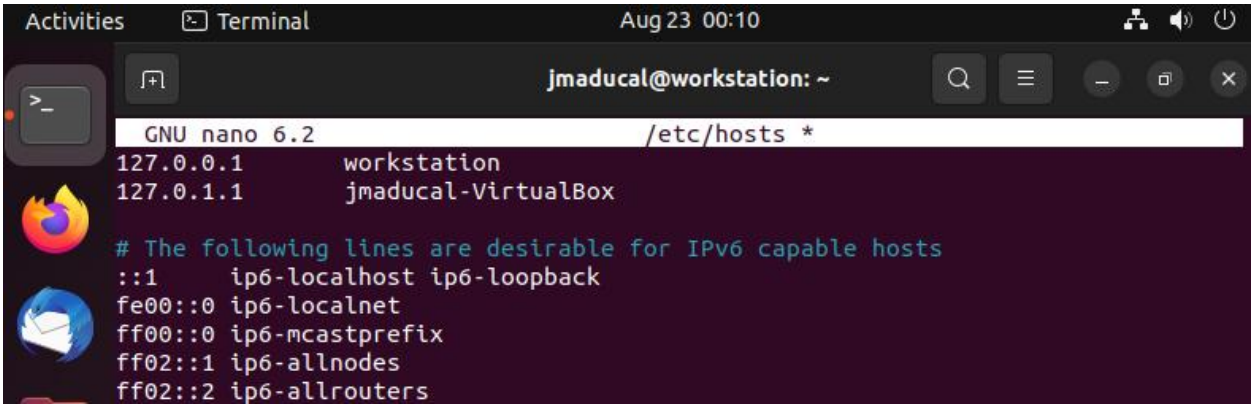
2.1 Type 127.0.0.1 server 1 for Server 1



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:

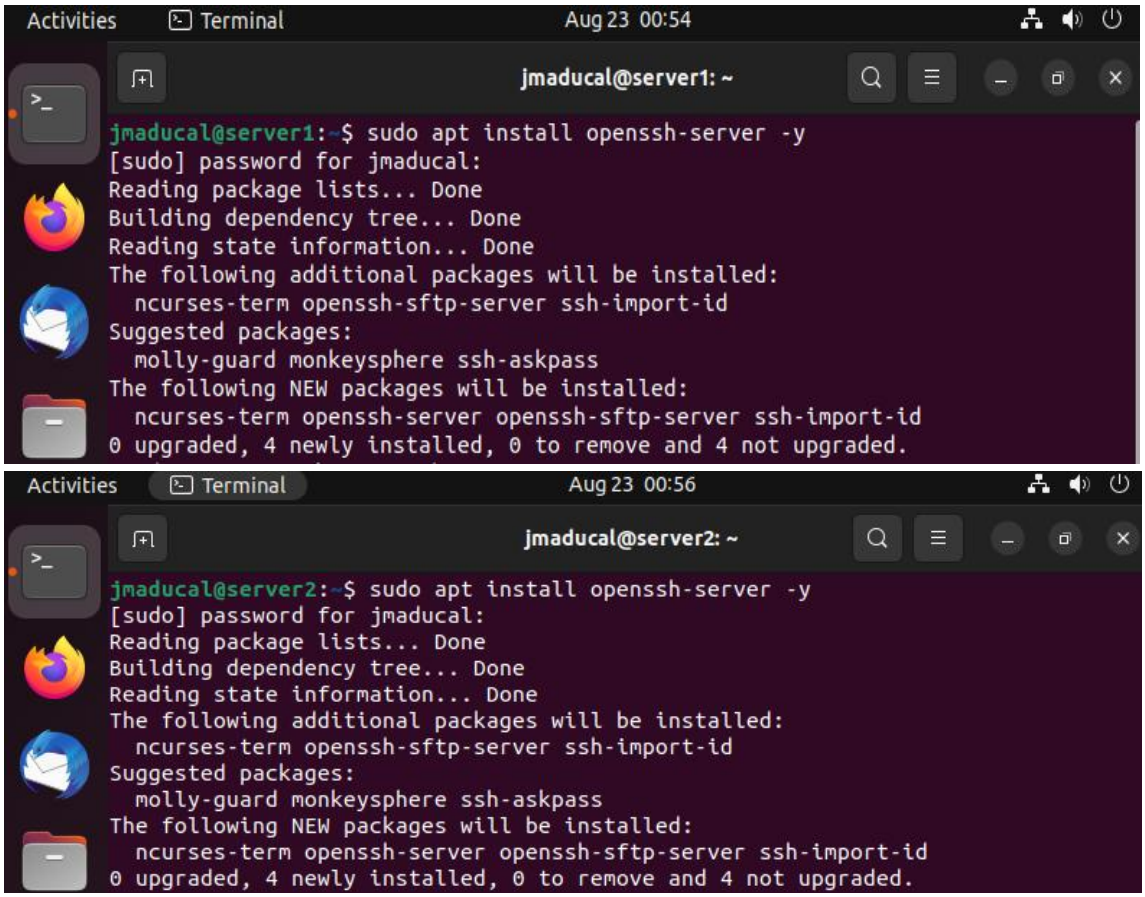
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.

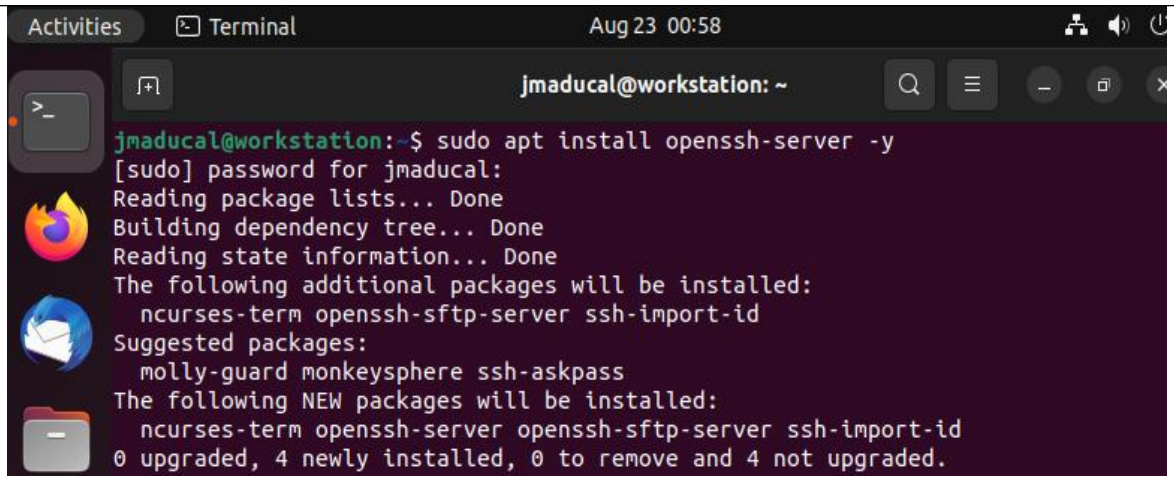
```
jmaducal@server1:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [274
 kB]
```

```
jmaducal@server2:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [274
 kB]
```

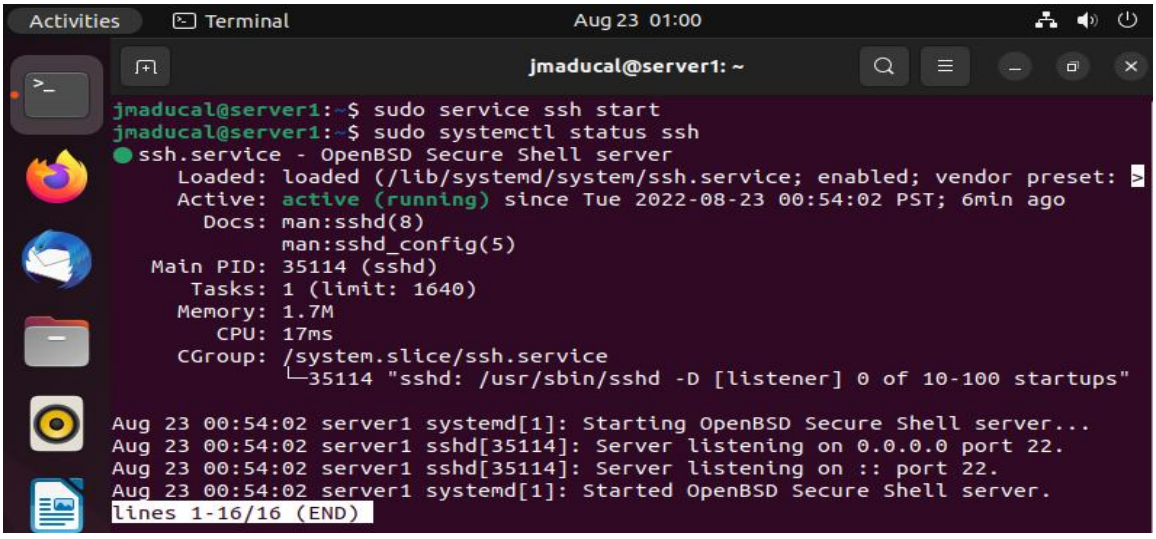
```
jmaducal@workstation:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [27
 kB]
```

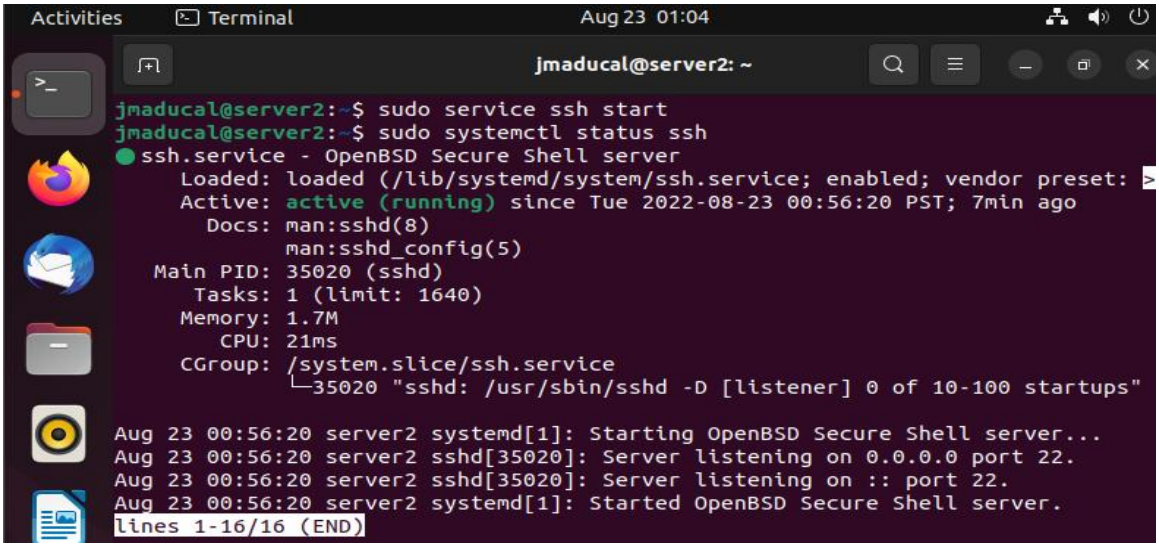
2. Install the SSH server using the command *sudo apt install openssh-server*.

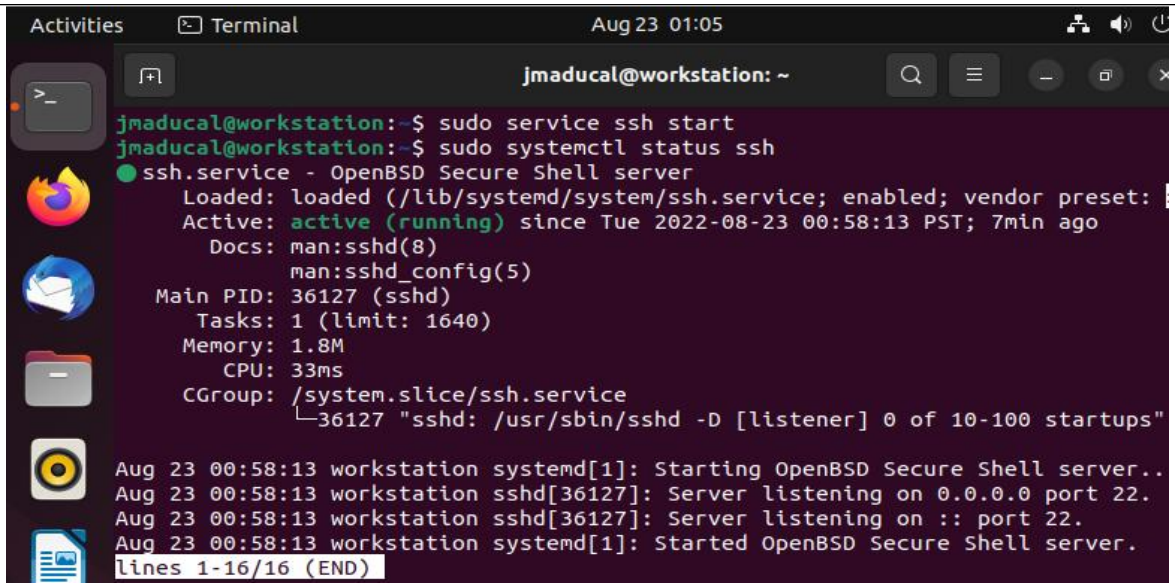




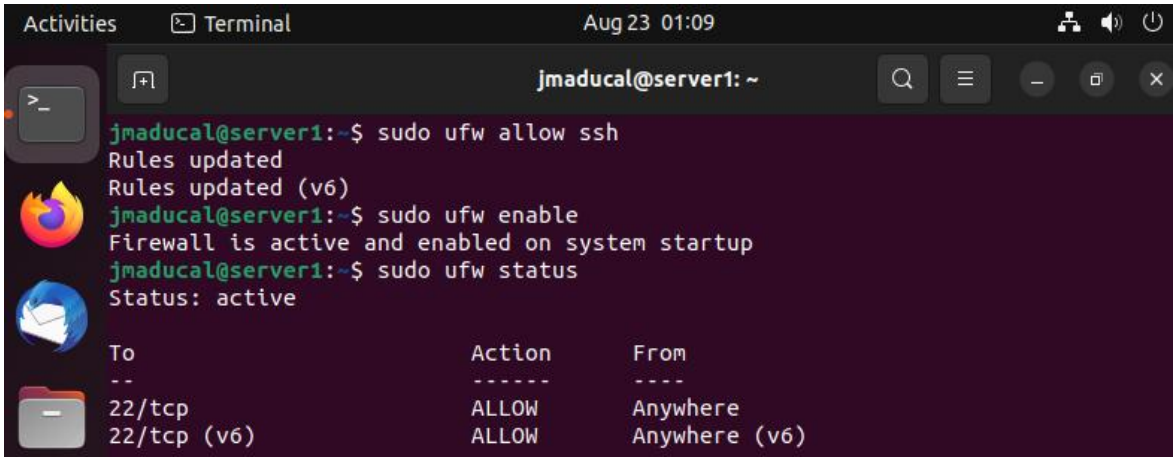
3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*

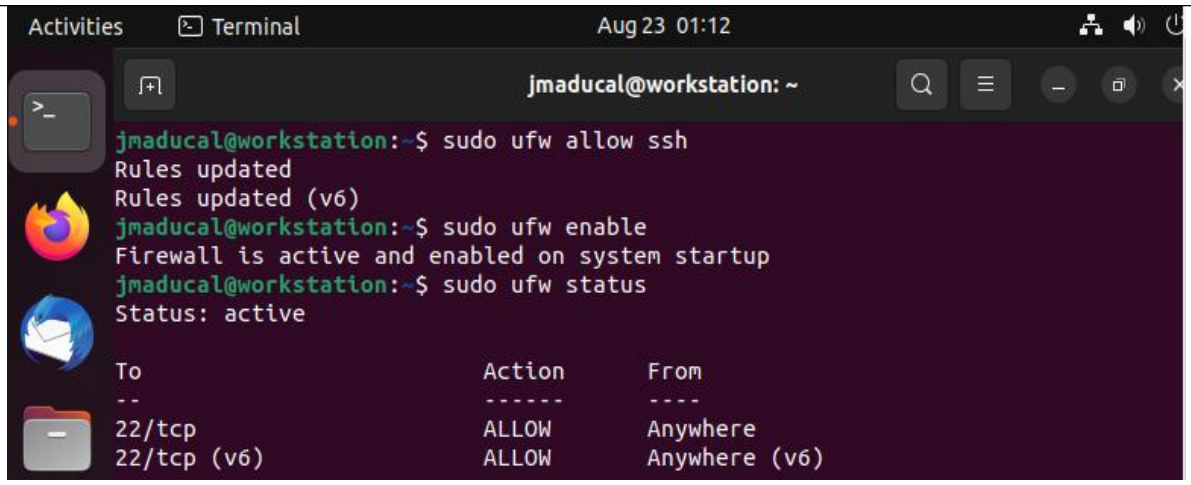






4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*

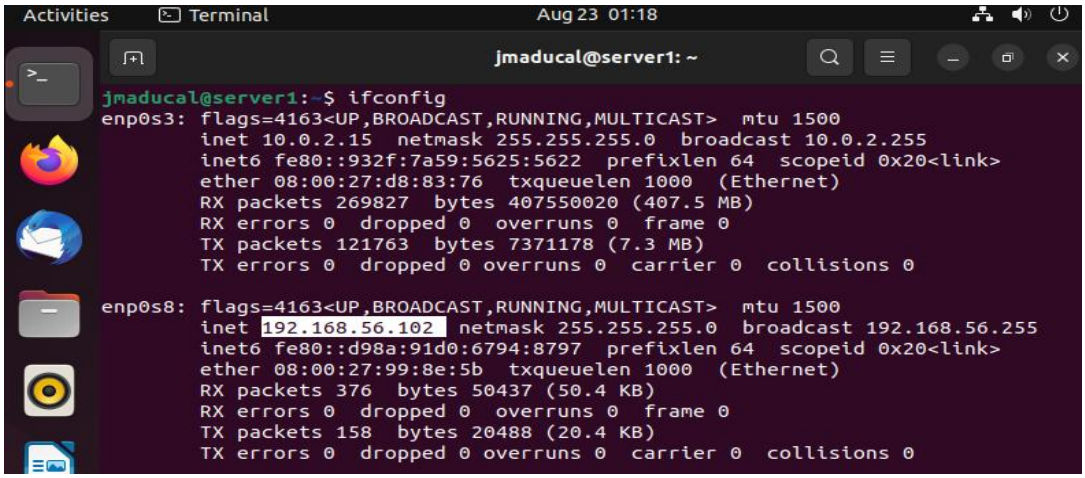




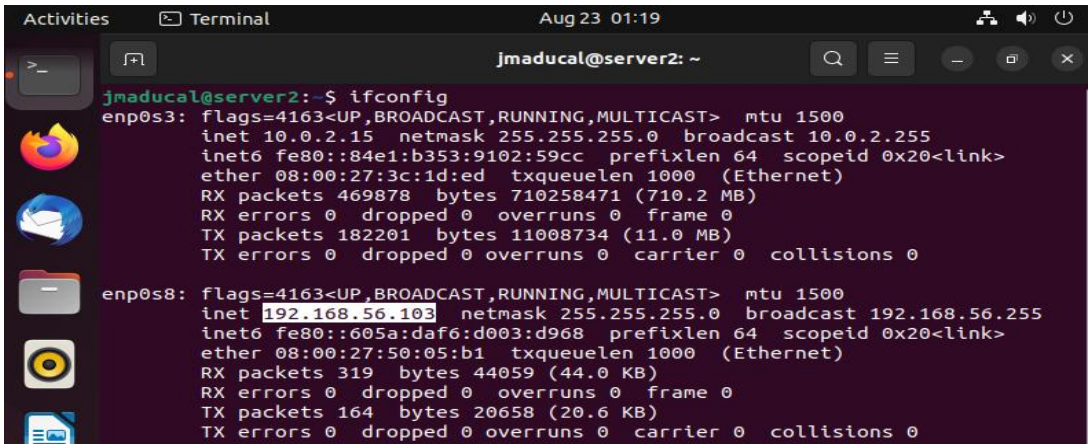
Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

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.

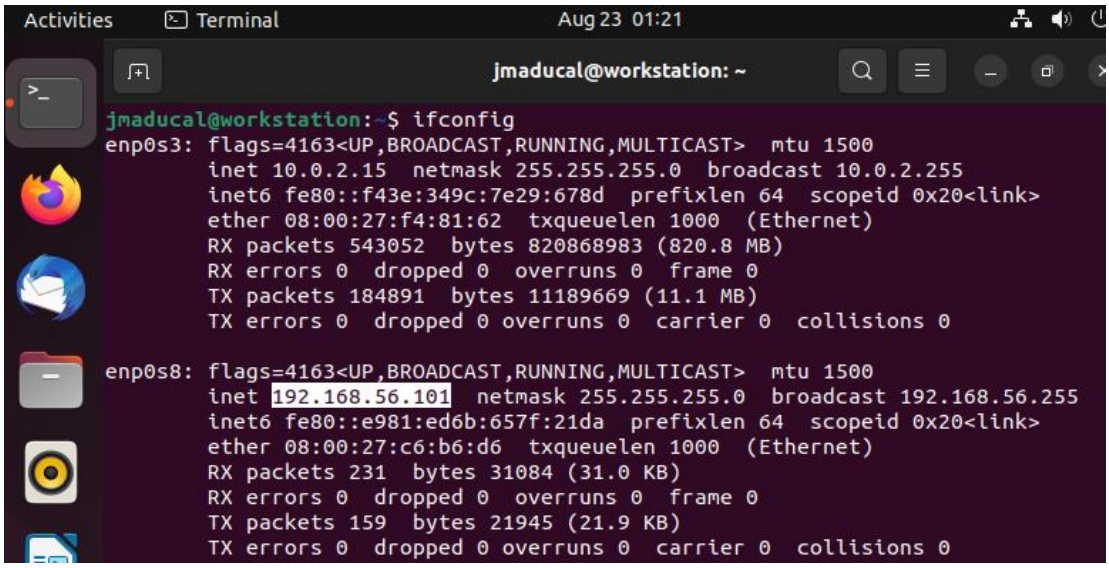
   1.1 Server 1 IP address: **192.168.56.102**



   1.2 Server 2 IP address: **192.168.56.103**

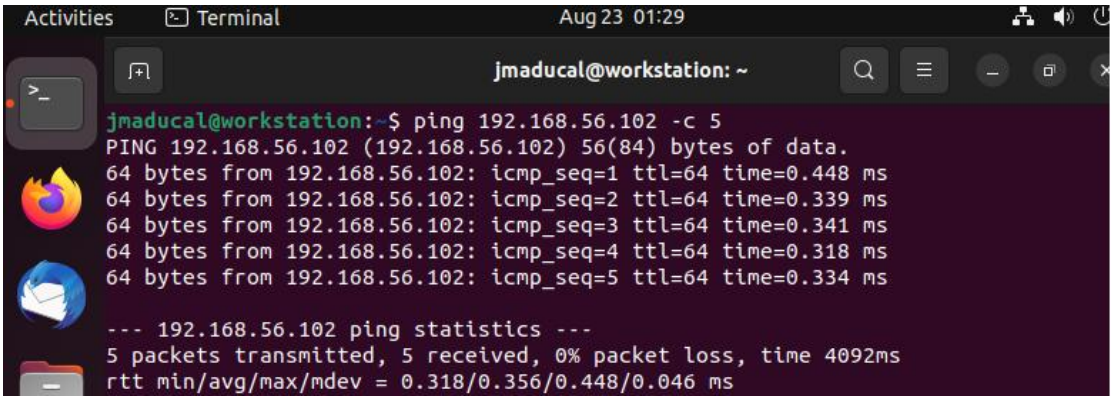


1.3 Workstation IP address: **192.168.56.101**

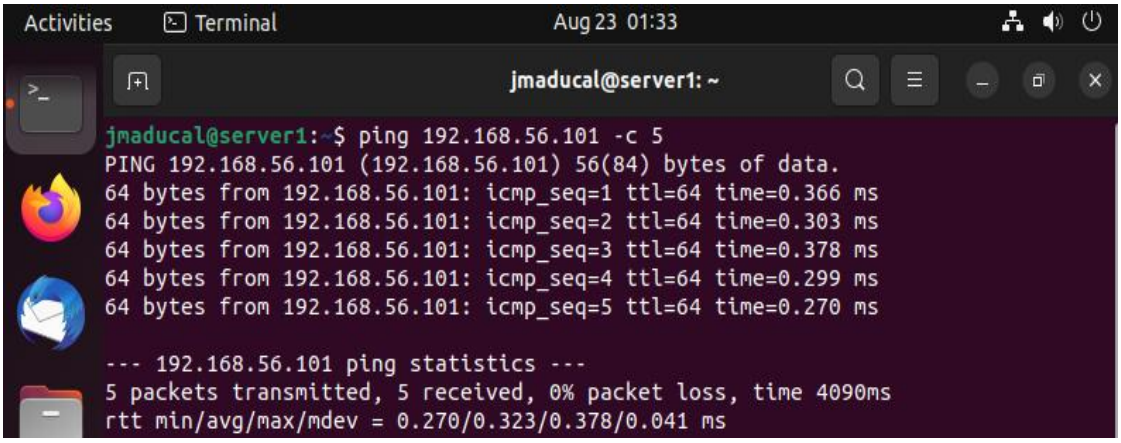


2. Make sure that they can ping each other.

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

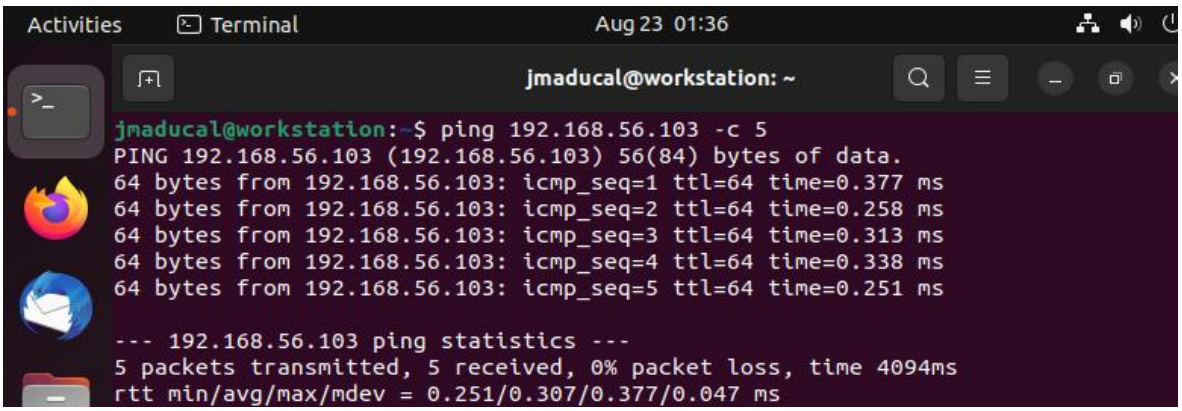


Ping from Local Machine 1 to Server 1

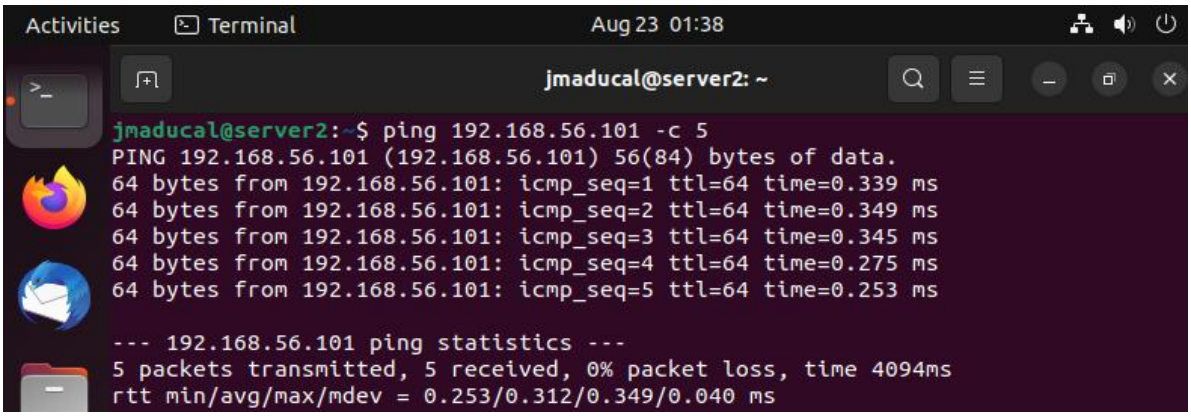


Ping from Server 1 to Local Machine 1

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

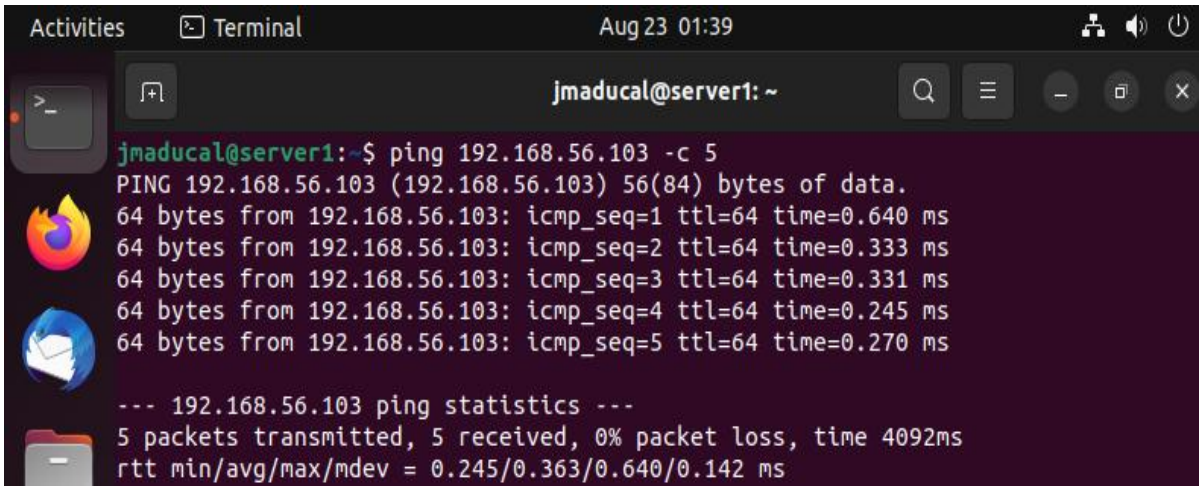


Ping from Local Machine 1 to Server 2

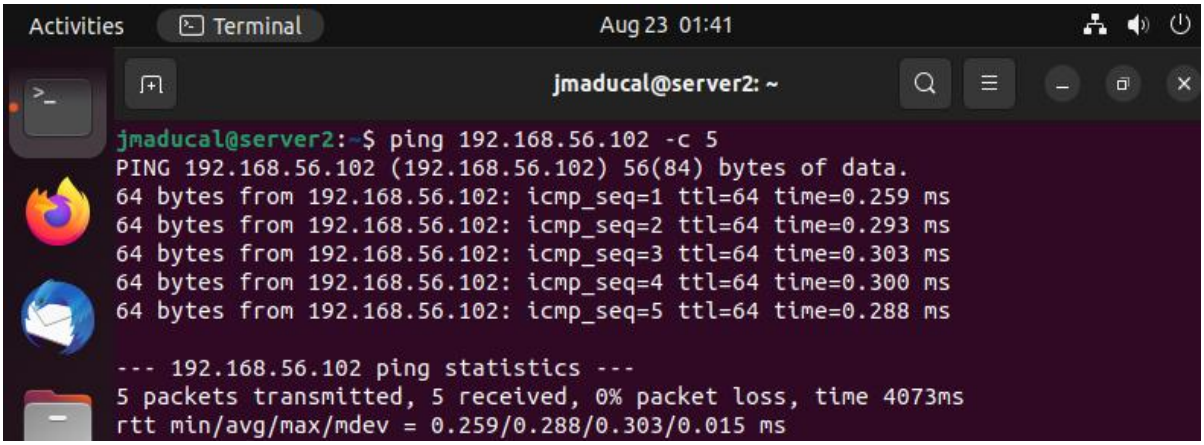


Ping from Server 2 to Local Machine 1

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



Ping from Server 1 to Server 2



Ping from Server 2 to Server 1

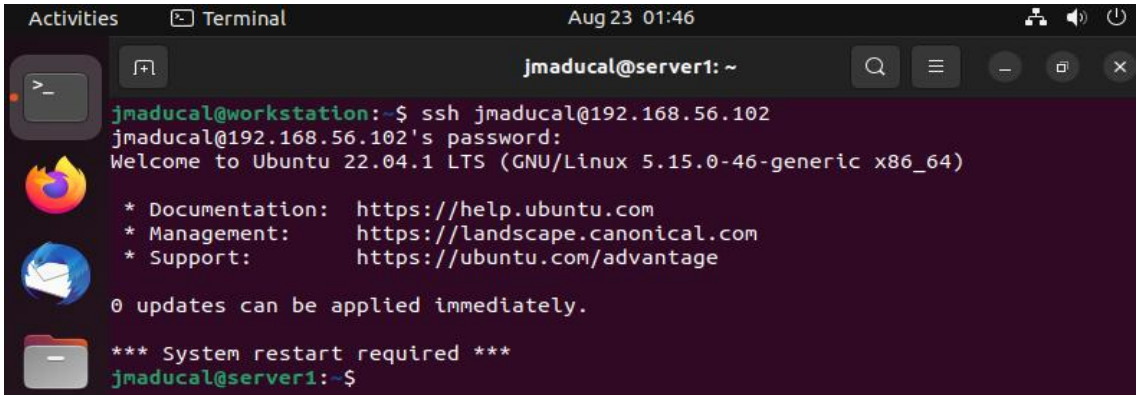
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

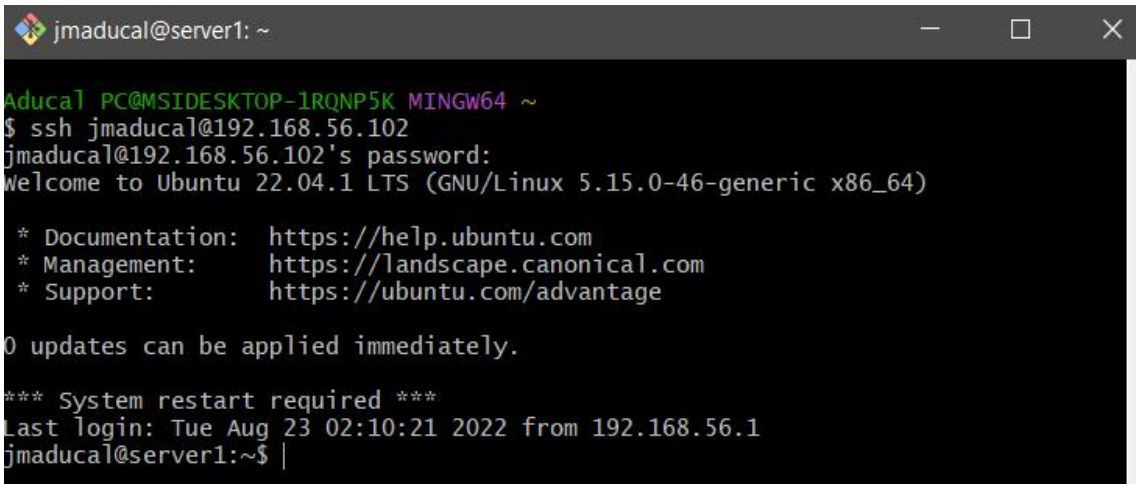
1. On the Local Machine, issue the following commands:

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

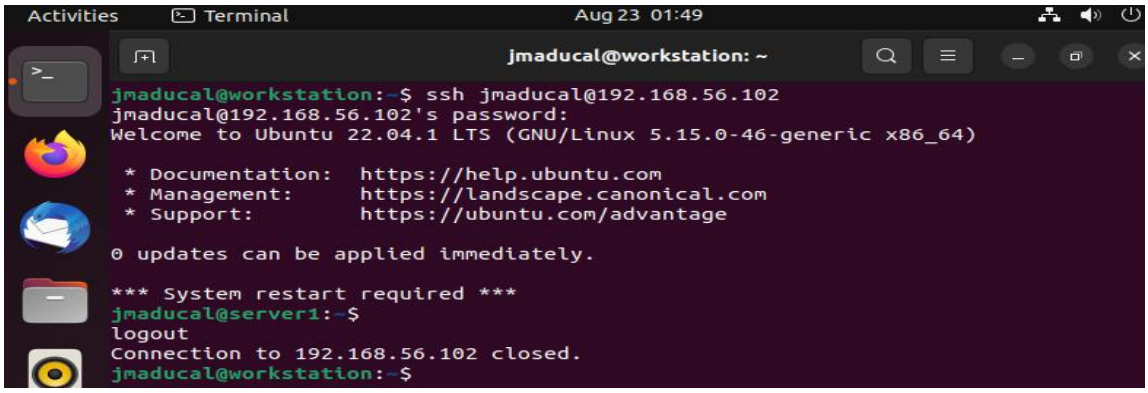
1.2 Enter the password for server 1 when prompted

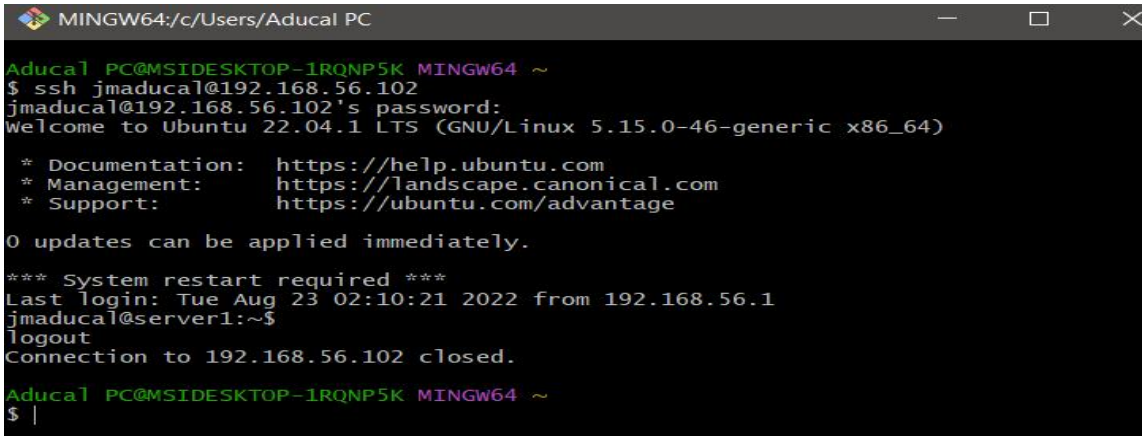
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*



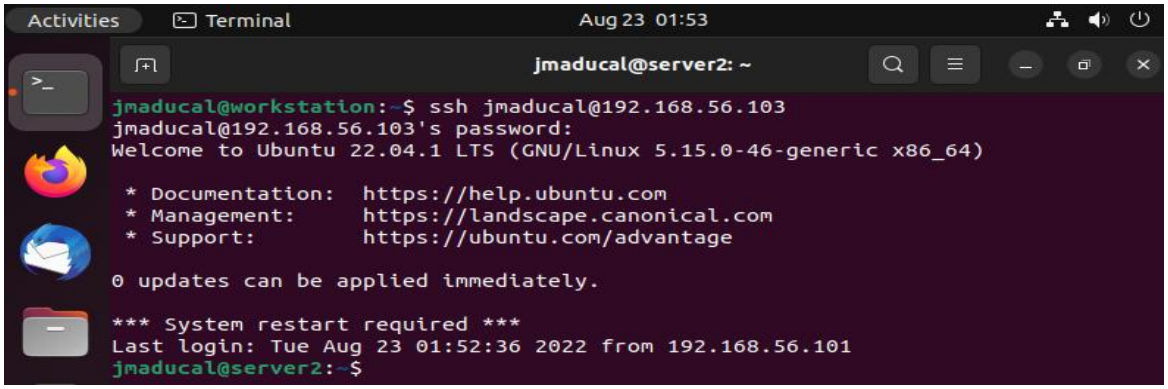


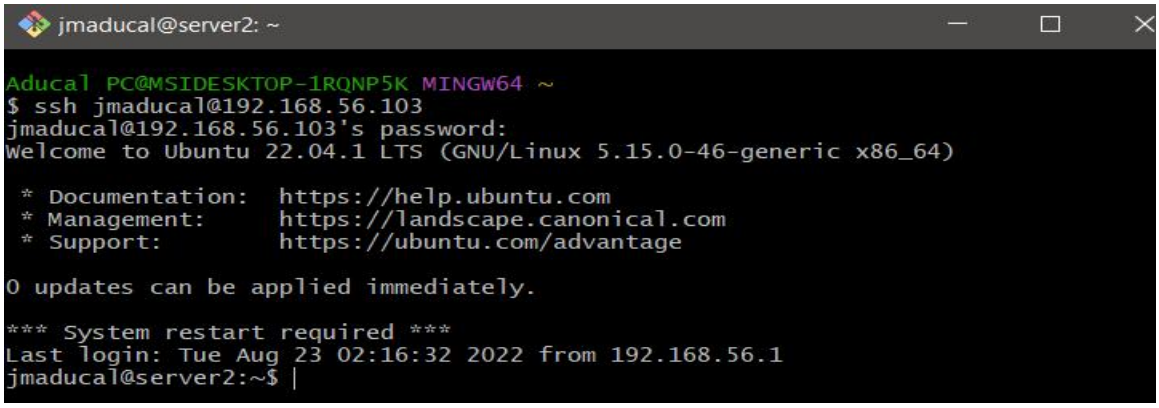
2. Logout of Server 1 by issuing the command *control + D.*

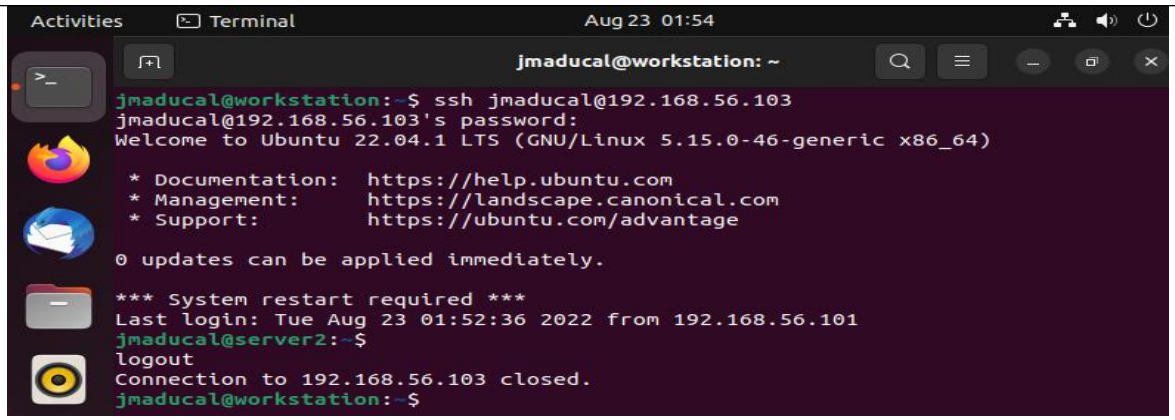


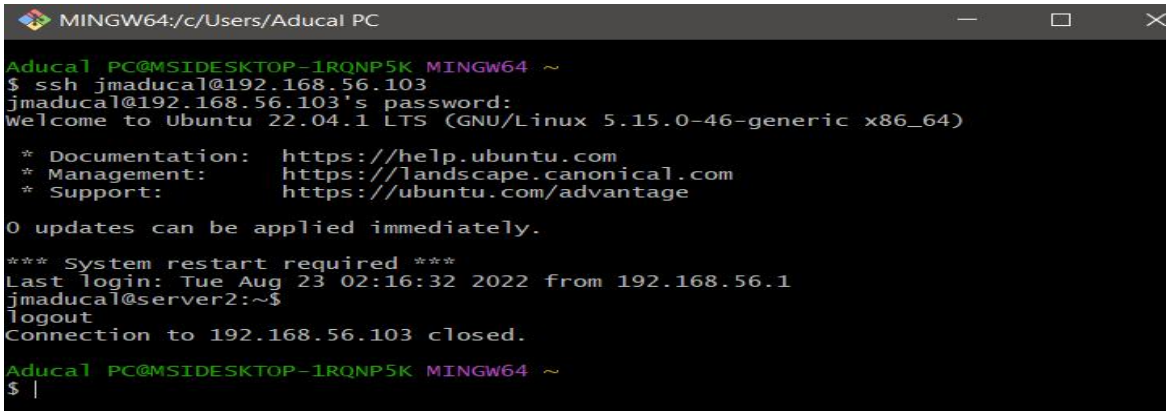


3. Do the same for Server 2.







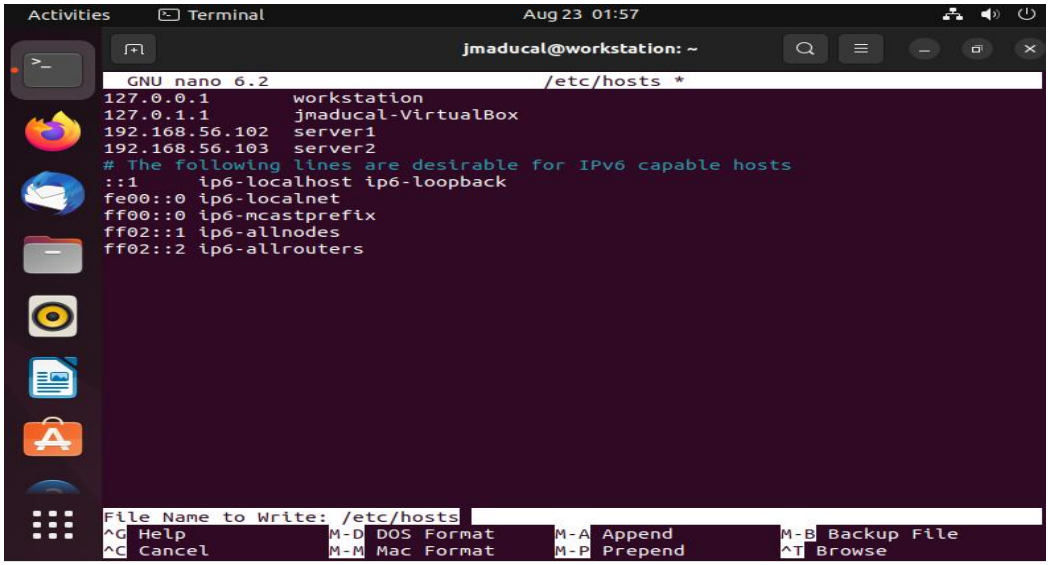


4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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

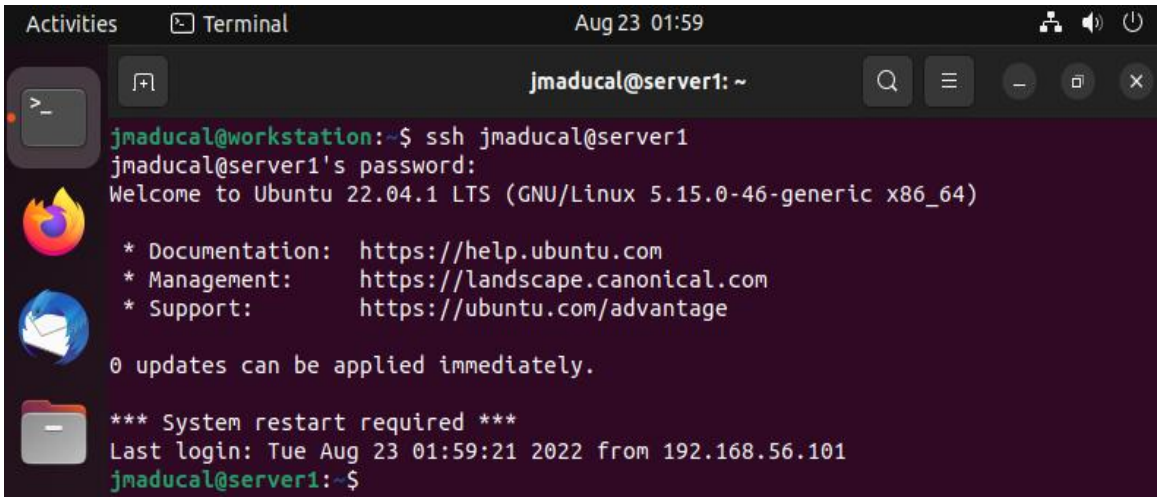
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.



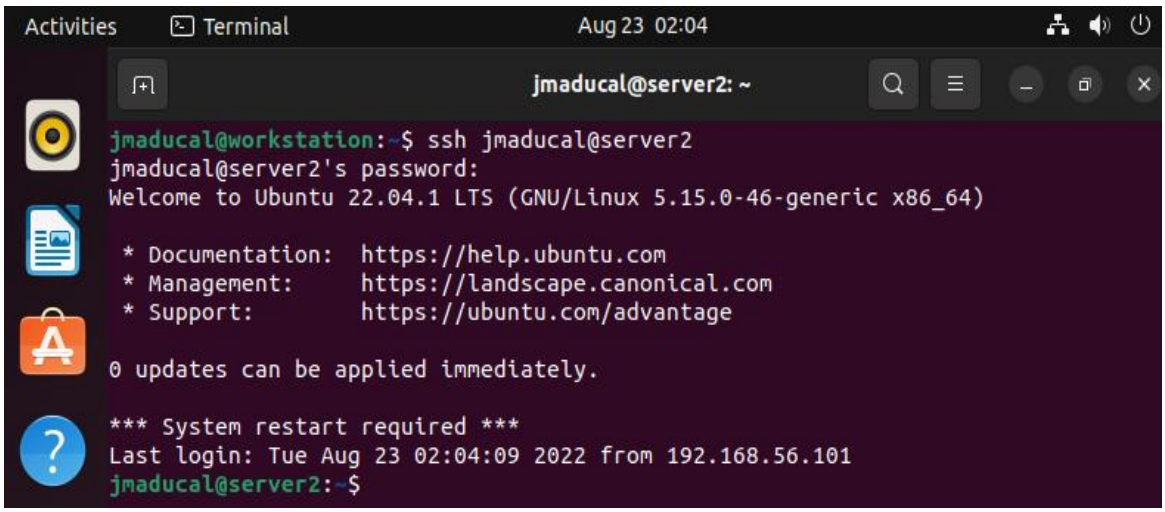
5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

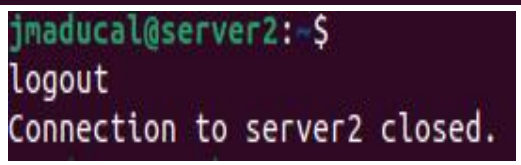
For Server 1:



CTRL + D to logout from server 1 and back to the Local Machine/Workstation.

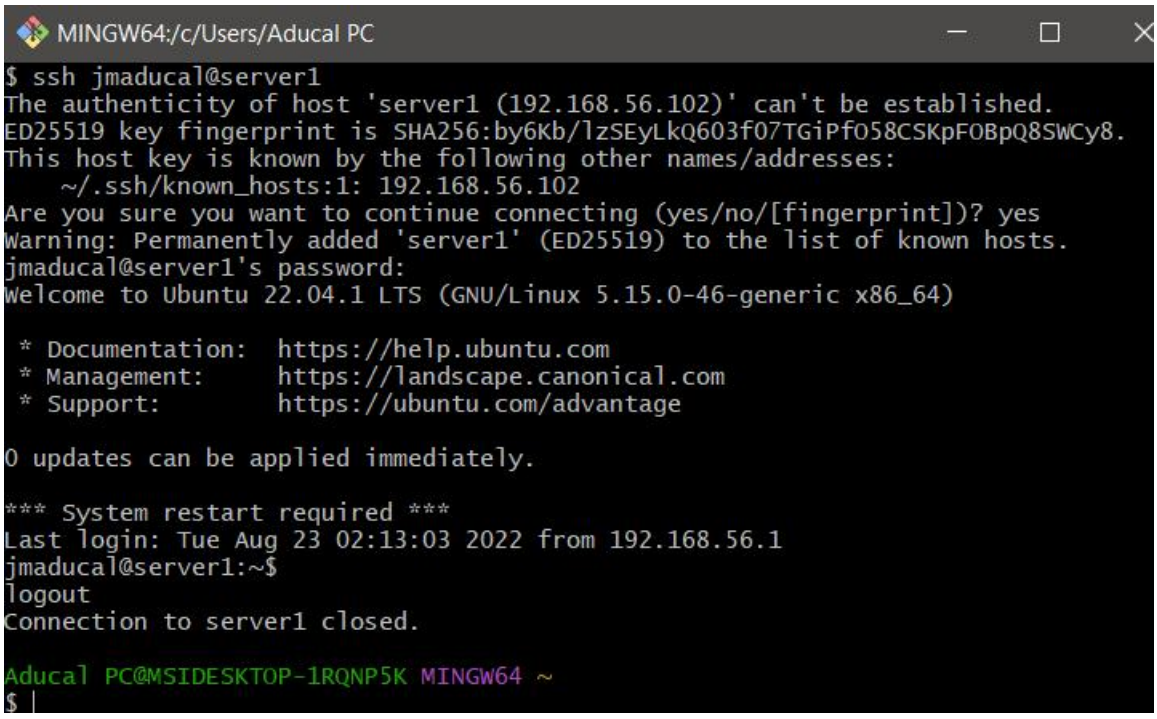
For Server 2:



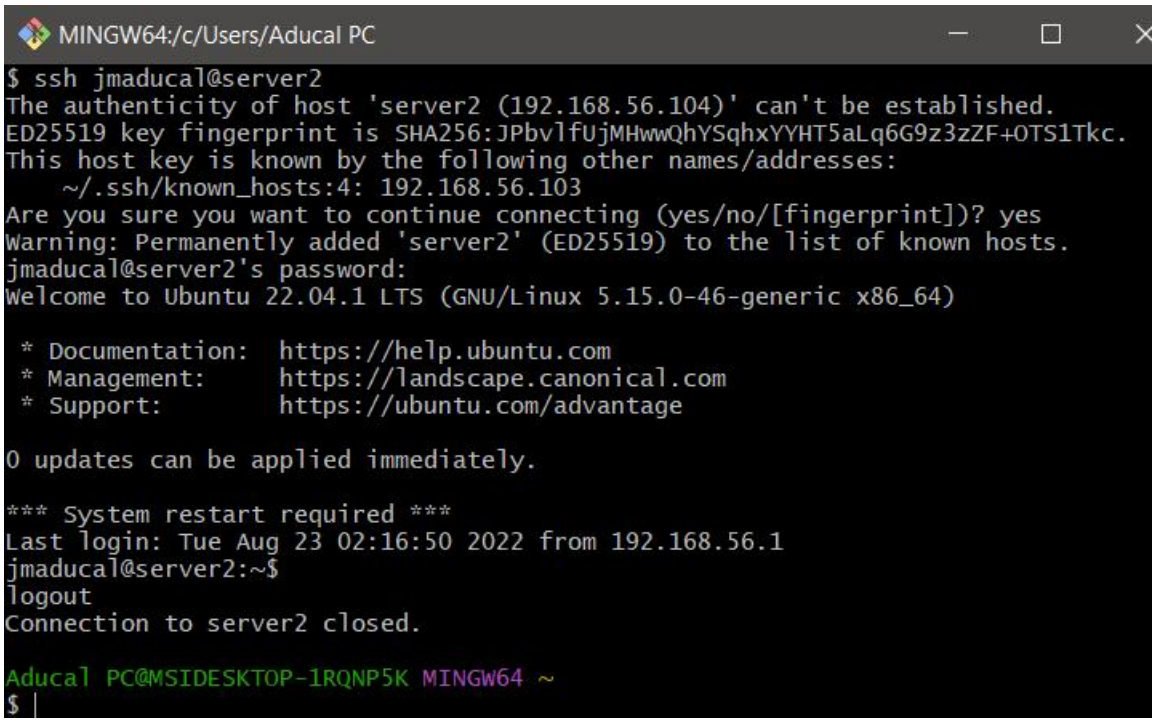


CTRL + D to logout from server 2 and back to the Local Machine/Workstation.

For Server 1:



For Server 2:



Reflections:

Answer the following:

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

We can do that by editing the hosts of the local machine issuing the command sudo nano /etc/hosts. Then provide or type the IP addresses of server 1 & 2 followed by their respective host name. Now we can use the host name instead of typing the IP address of the servers.

2. How secured is SSH?

SSH connections have mostly been used to secure different types of communications between a local machine and a remote host, including:

- Secure remote access to resources
- Remote execution of commands
- Delivery of software patches and updates
- Interactive and automated file transfers

In addition to creating a secure channel between local and remote computers, SSH protocol is used for managing critical corporate infrastructure such as routers, server hardware, virtualization platforms, and operating systems.

SSH keys are used to automate access to servers and often are used in scripts, backup systems and configuration management tools. Because of their design that allows connectivity across organizational boundaries, SSH keys provide single sign-on (SSO) capabilities allowing users to move between their accounts without typing a password each time.

Reference:

W. Altaqi. "The What, Why & How of SSH Protocol – Keyfactor." https://www.keyfactor.com/blog/ssh-protocol *(accessed: Aug. 23, 2022).*

Honor Pledge:

"I affirm that I shall not give or receive any unauthorized help on this activity and all the work shall be my own"