Name: Emmanuel G. Pascua	Date Performed: August 16, 2022
Course/Section: CPE 232/CPE31S22	Date Submitted: August 20, 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem 2022 - 2023

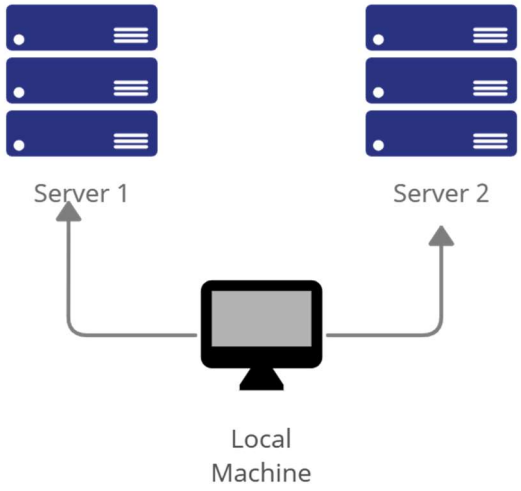
Activity 1: Configure Network using Virtual Machines

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:**

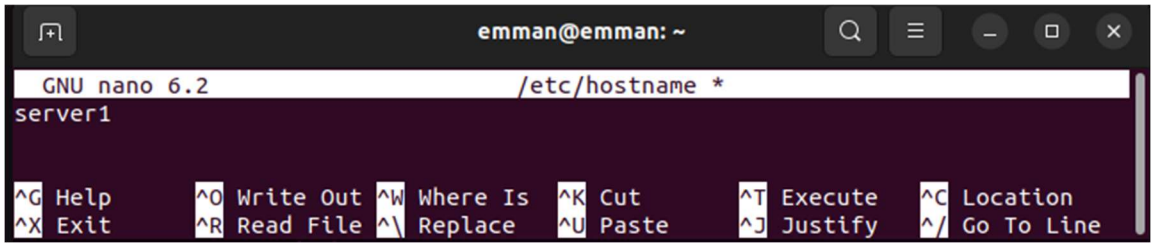
Network Topology:

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*).

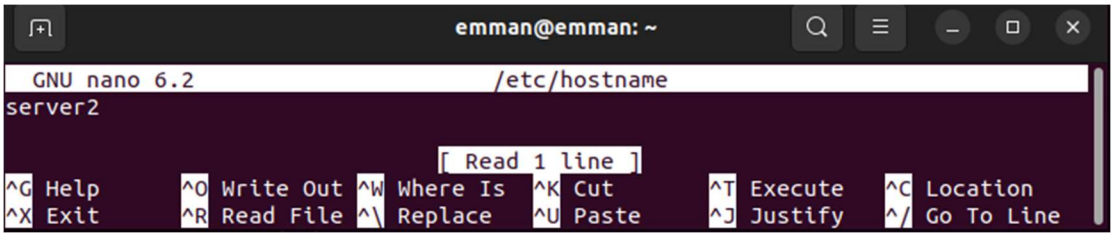


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

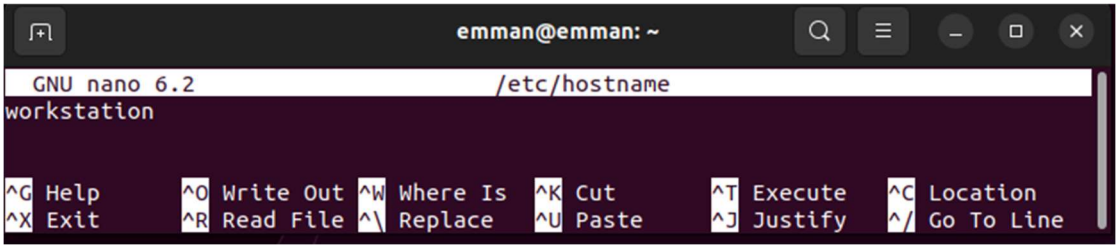
1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1



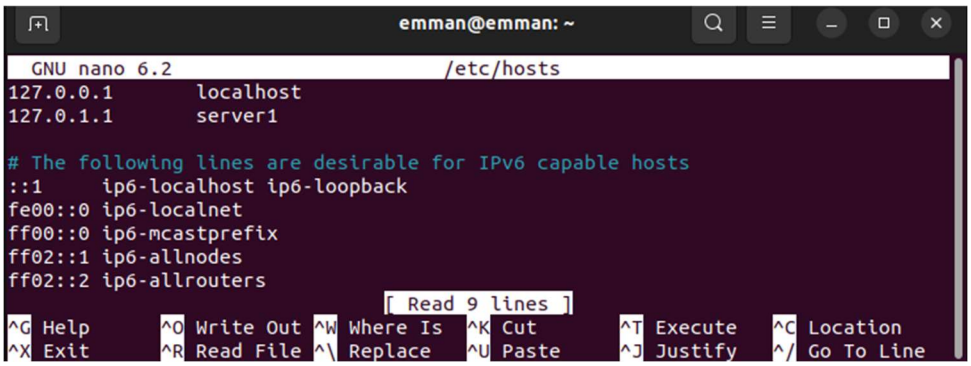
1.2 Use server2 for Server 2



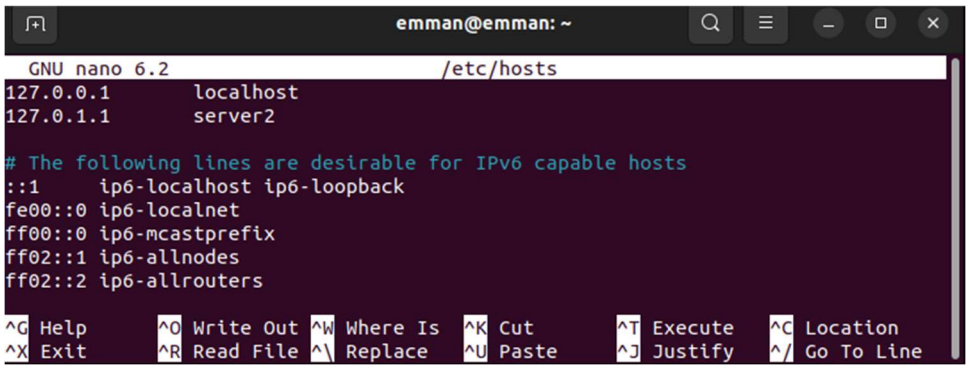
1.3 Use workstation for the Local Machine



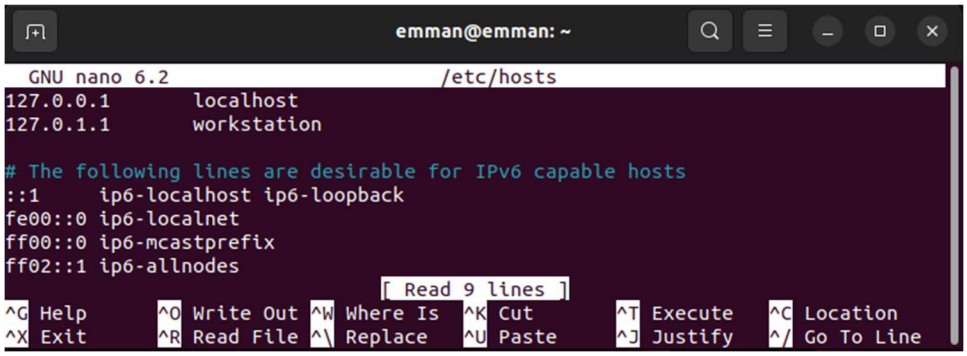
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2

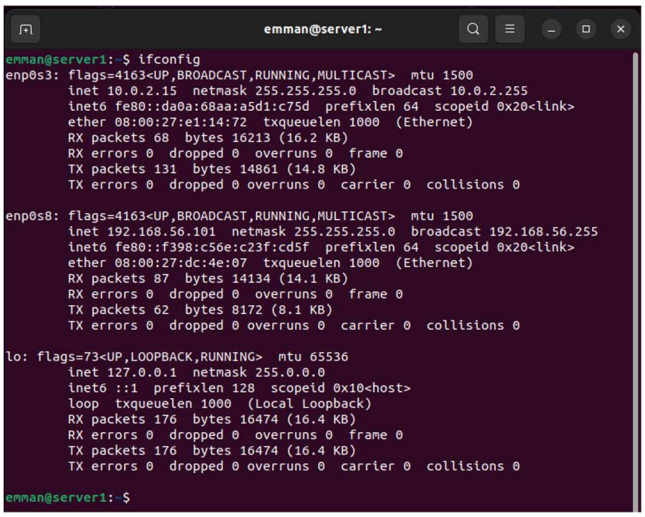


2.3 Type 127.0.0.1 workstation for the Local Machine

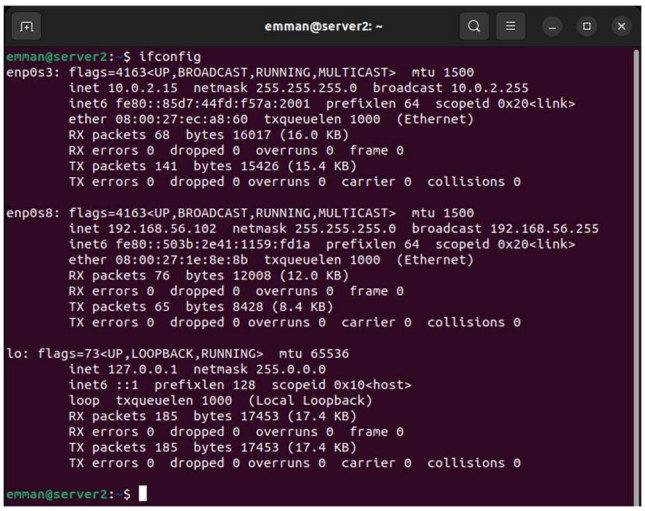


After Rebooting:

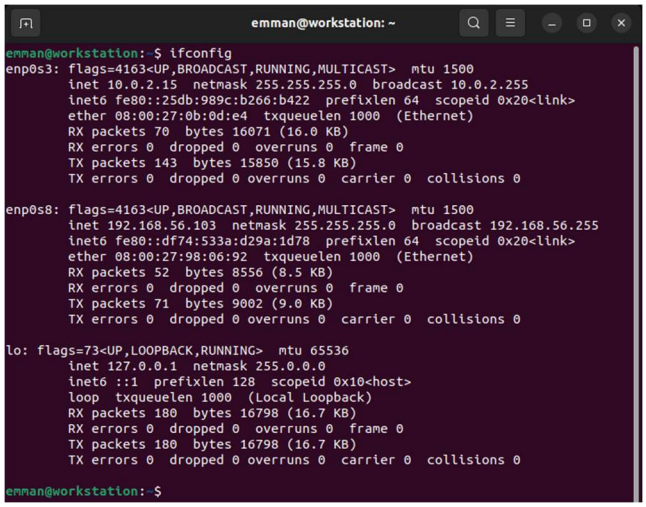
Server 1:



Server 2:



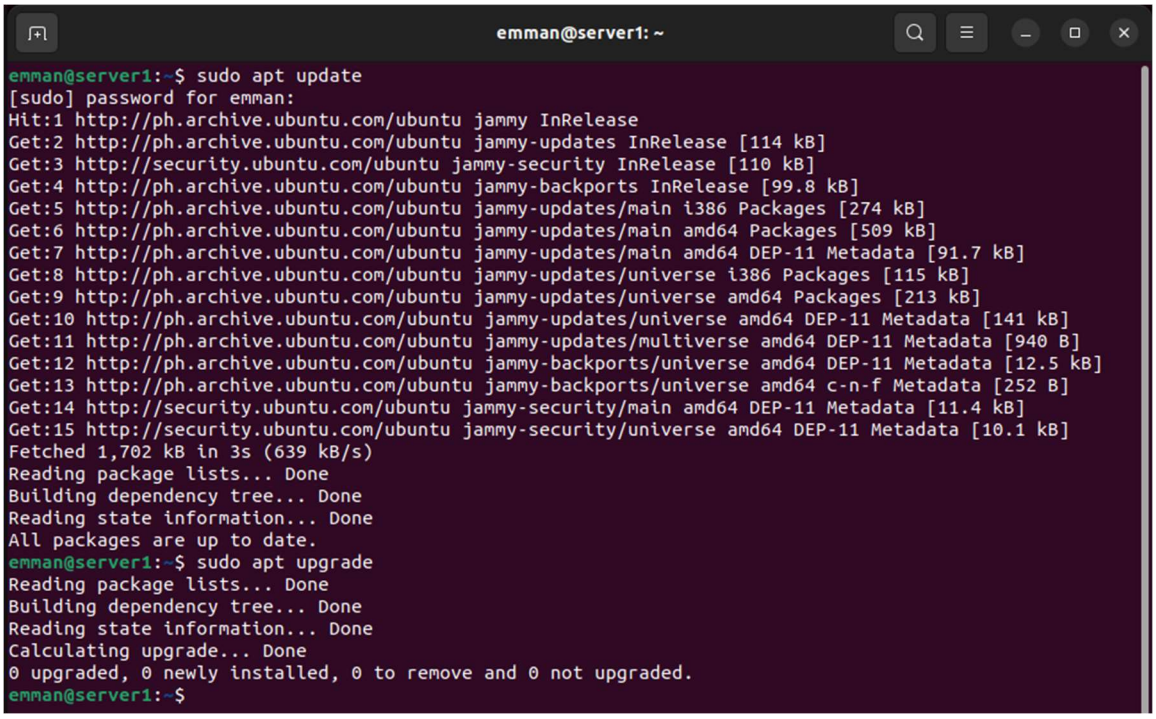
Workstation:



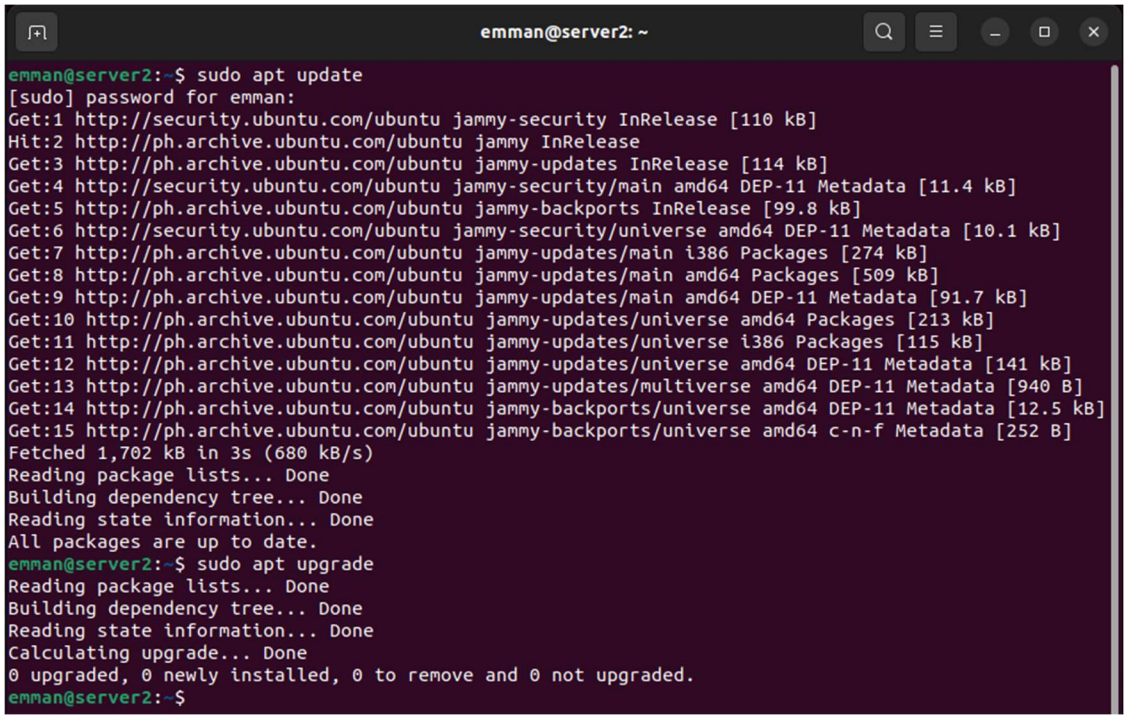
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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

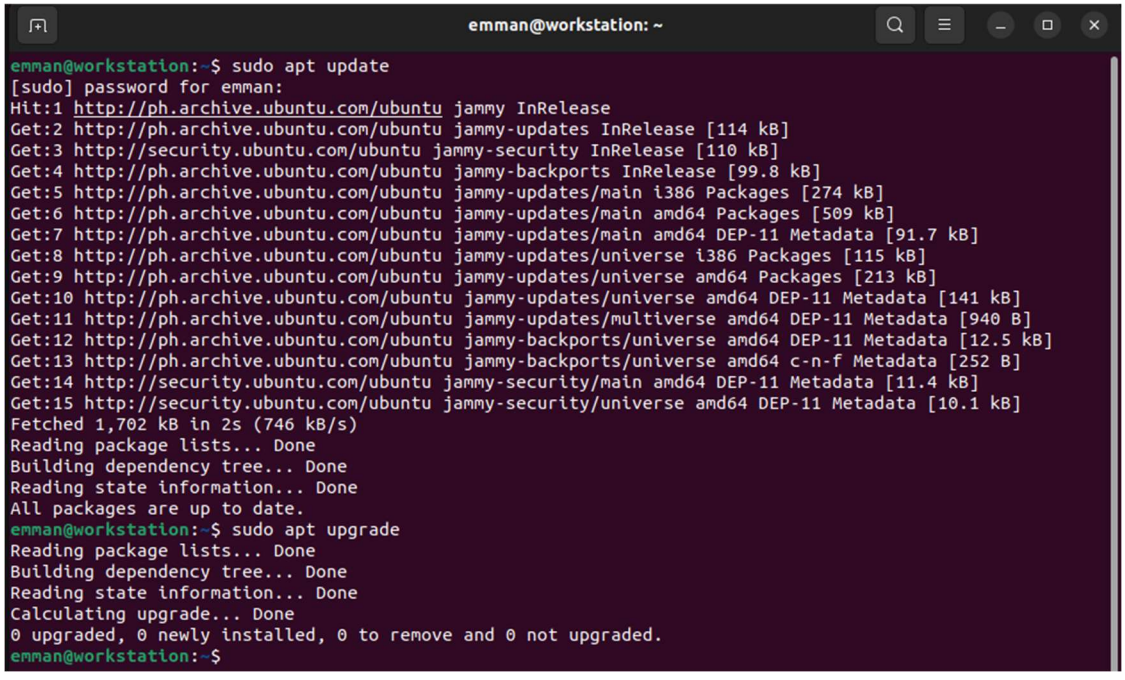
Server 1:



Server 2:

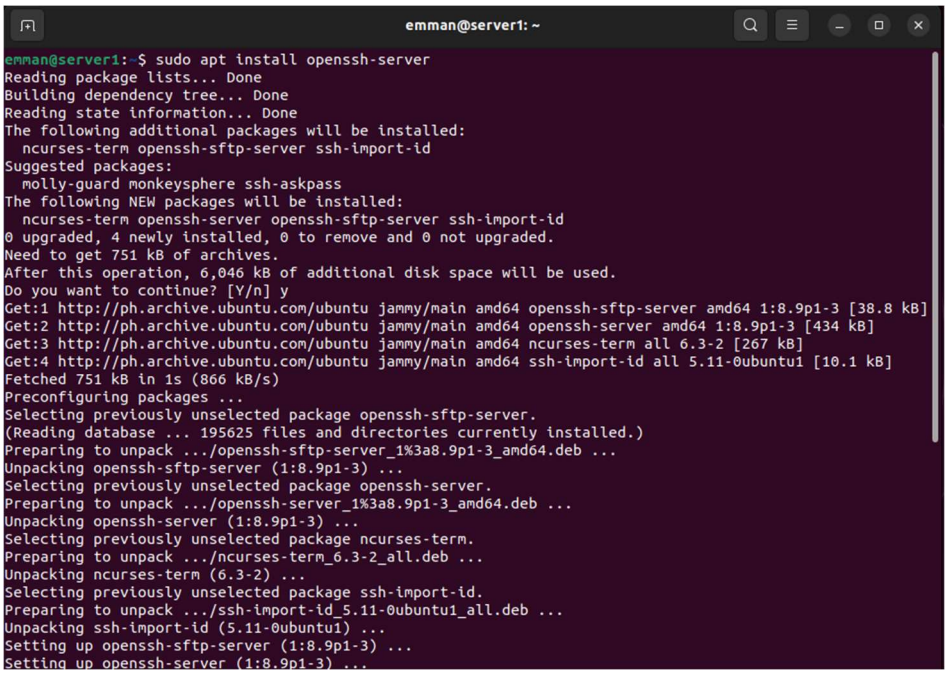


Workstation:

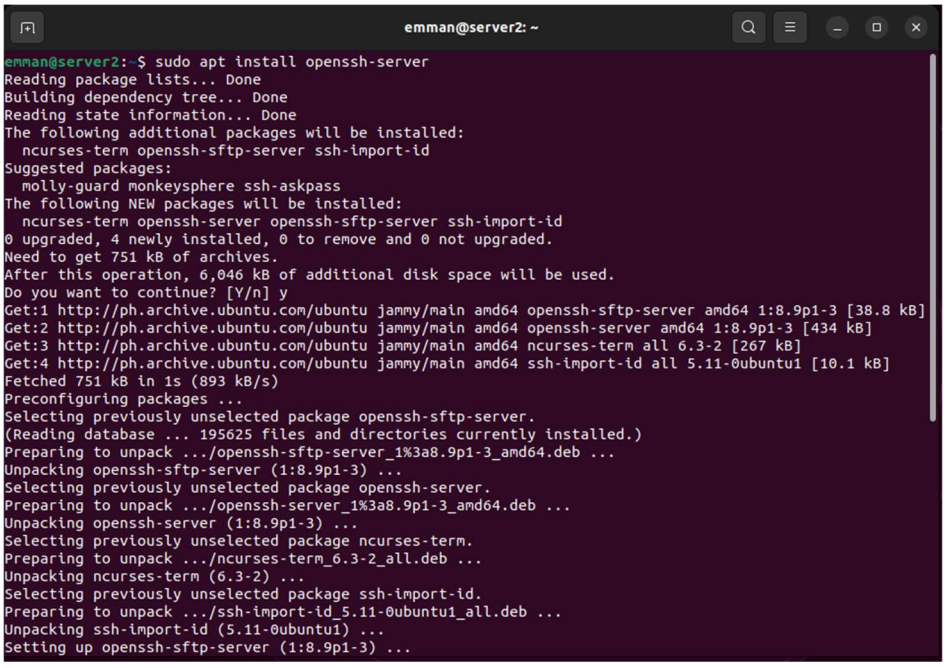


2. Install the SSH server using the command *sudo apt install openssh-server*.

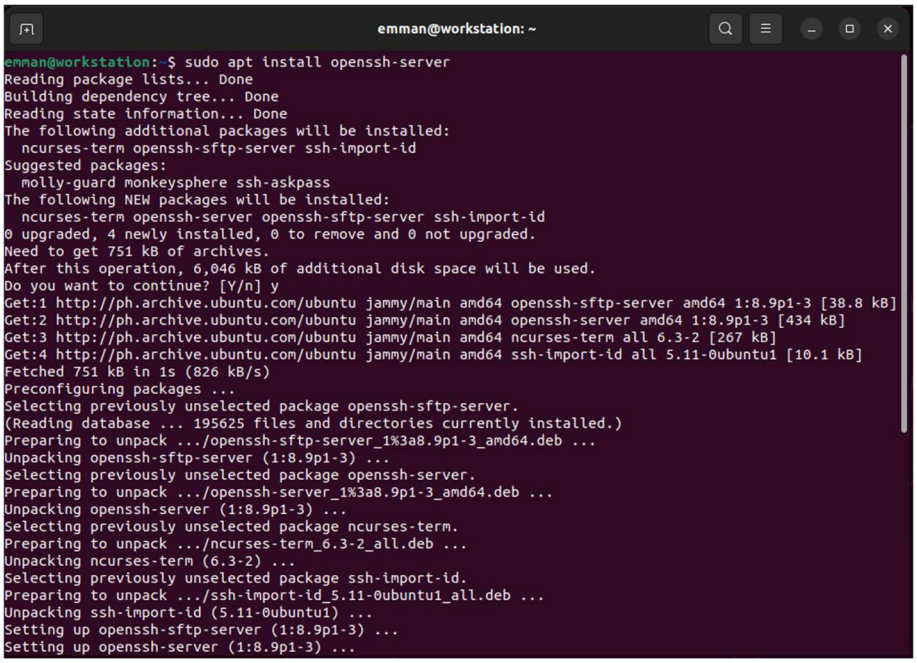
Server 1:



Server 2:



Workstation:

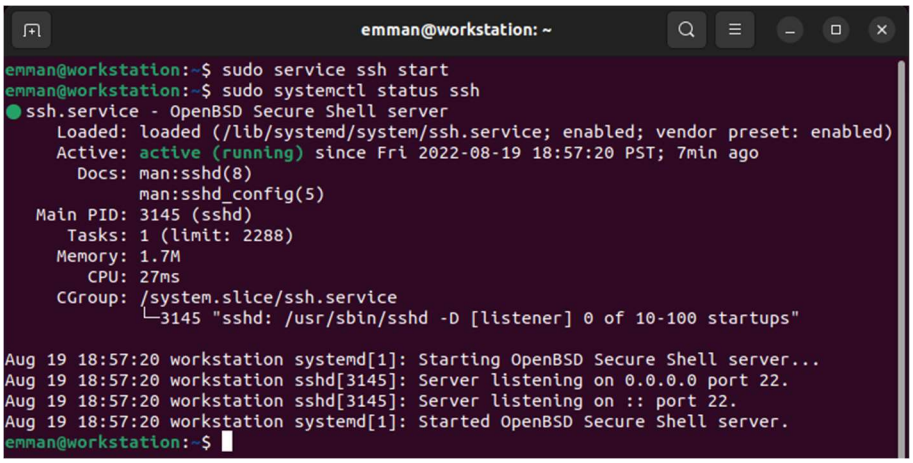


3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*
   3.2 *sudo systemctl status ssh*

Server 1:

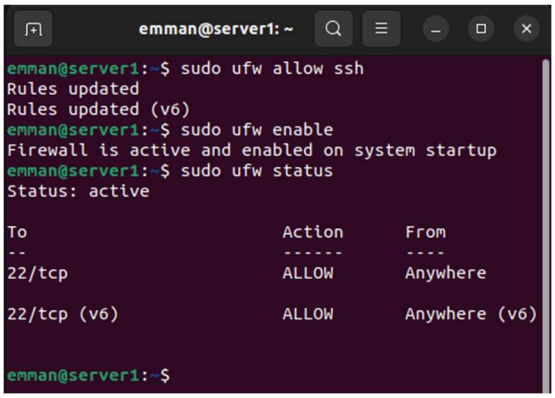
Server 2:

Workstation:

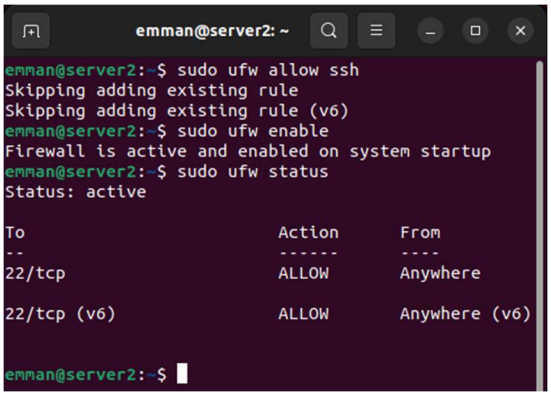


4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*
   4.2 *sudo ufw enable*
   4.3 *sudo ufw status*

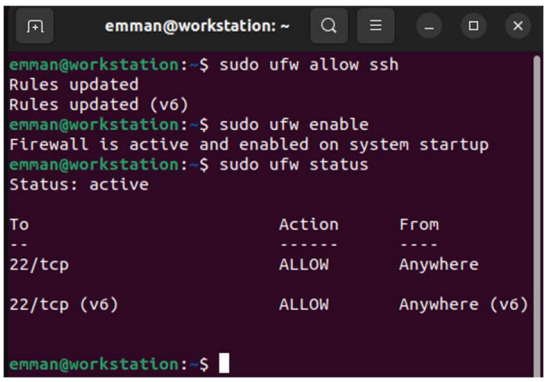
Server 1:



Server 2:



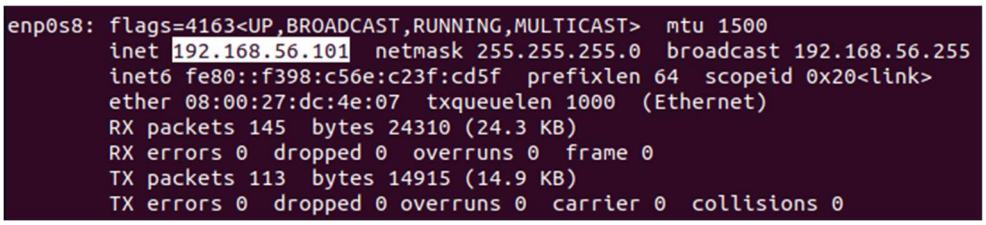
Workstation:



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.___
   1.2 Server 2 IP address: 192.168.56.___
   1.3 Server 3 IP address: 192.168.56.___

Server 1:



Server 2:

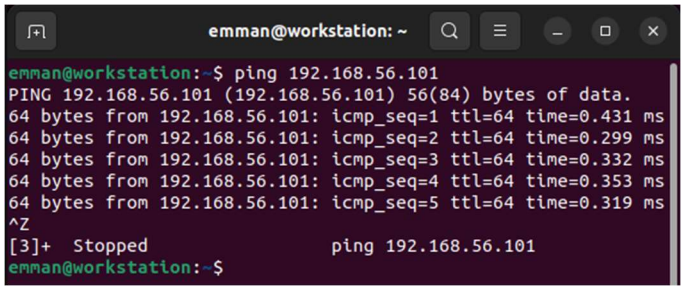
```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::503b:2e41:1159:fd1a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1e:8e:8b  txqueuelen 1000  (Ethernet)
        RX packets 132  bytes 22458 (22.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 15522 (15.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Workstation:

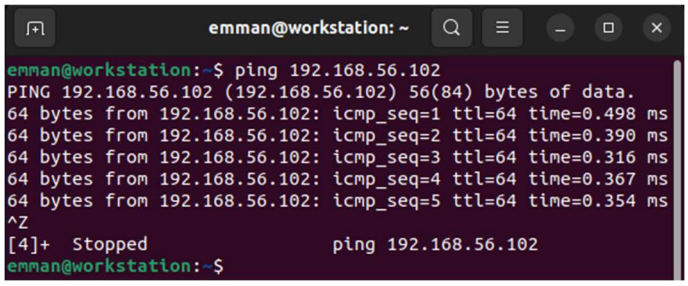
```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.103  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::df74:533a:d29a:1d78  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:98:06:92  txqueuelen 1000  (Ethernet)
        RX packets 94  bytes 17720 (17.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 129  bytes 16694 (16.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Make sure that they can ping each other.

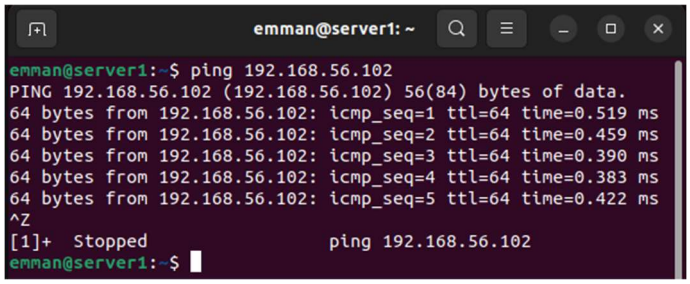
2.1 Connectivity test for Local Machine 1 to Server 1: ☒ Successful ☐ Not Successful



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



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



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

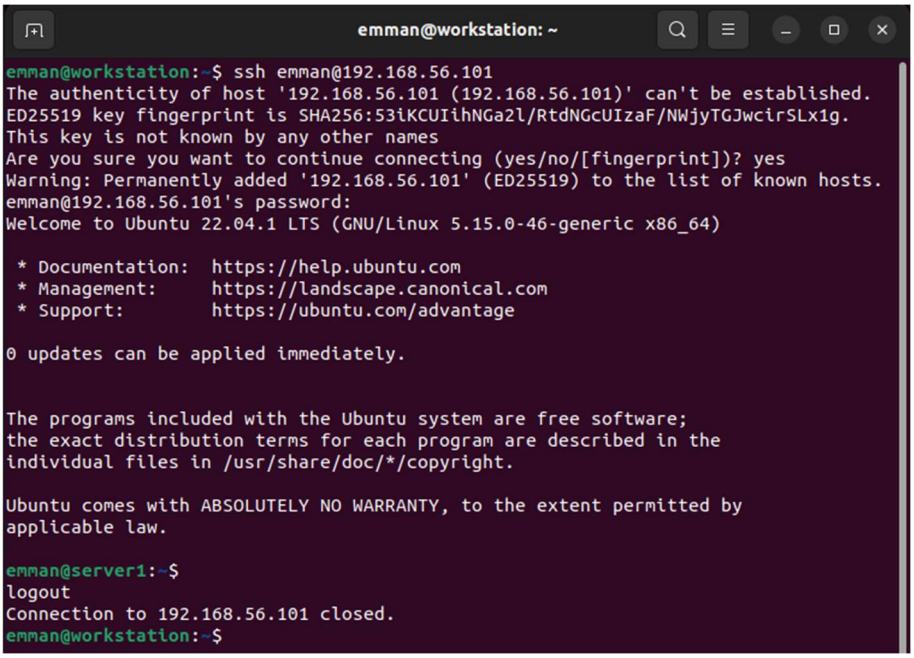
1. On the Local Machine, issue the following commands:

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

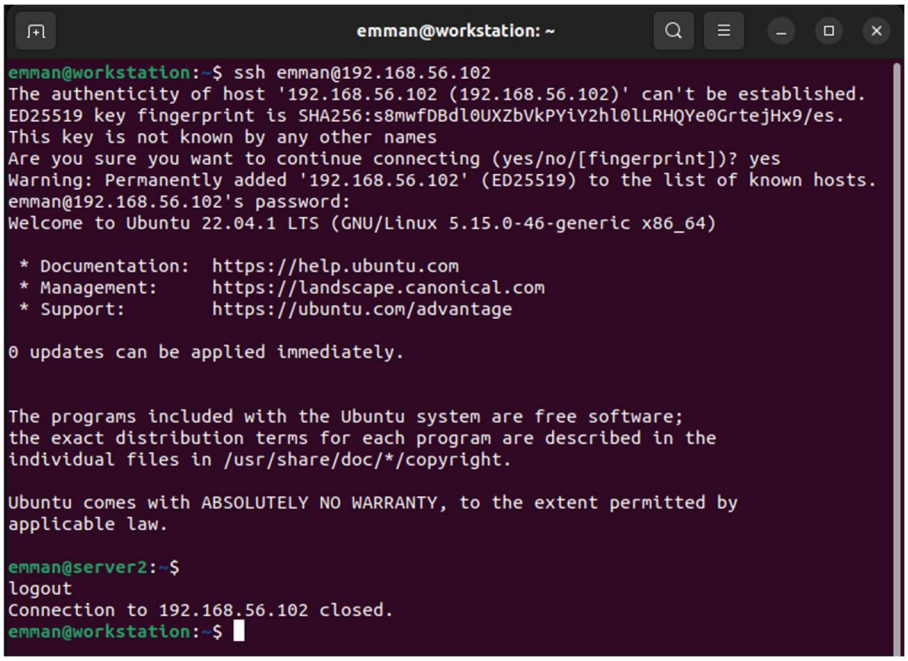
1.2 Enter the password for server 1 when prompted

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.

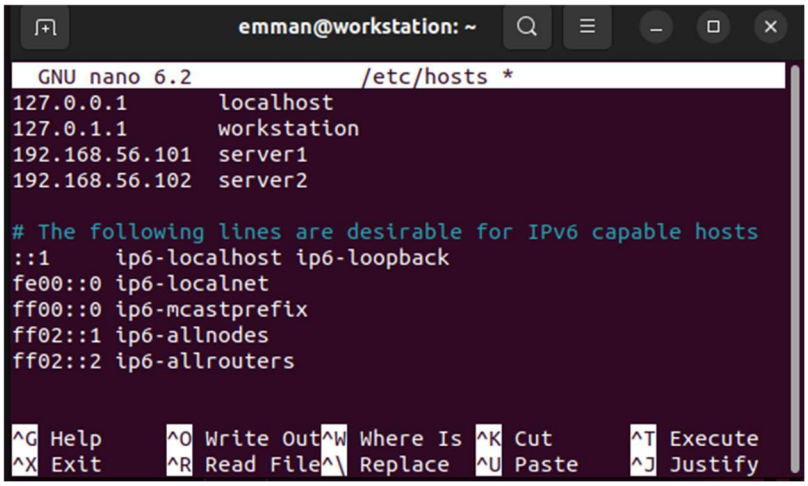


4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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

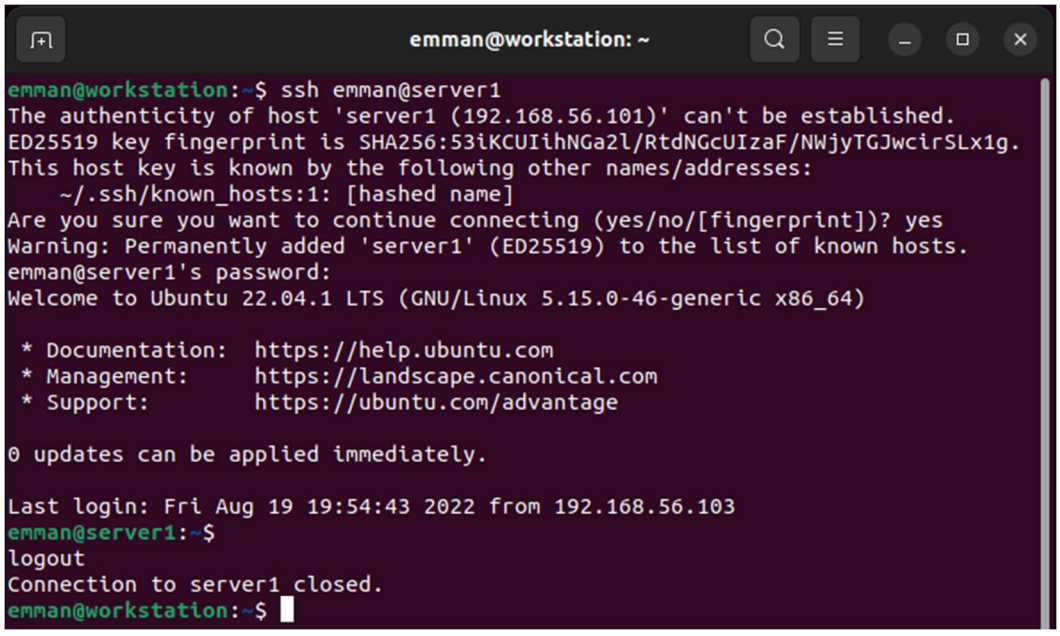
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

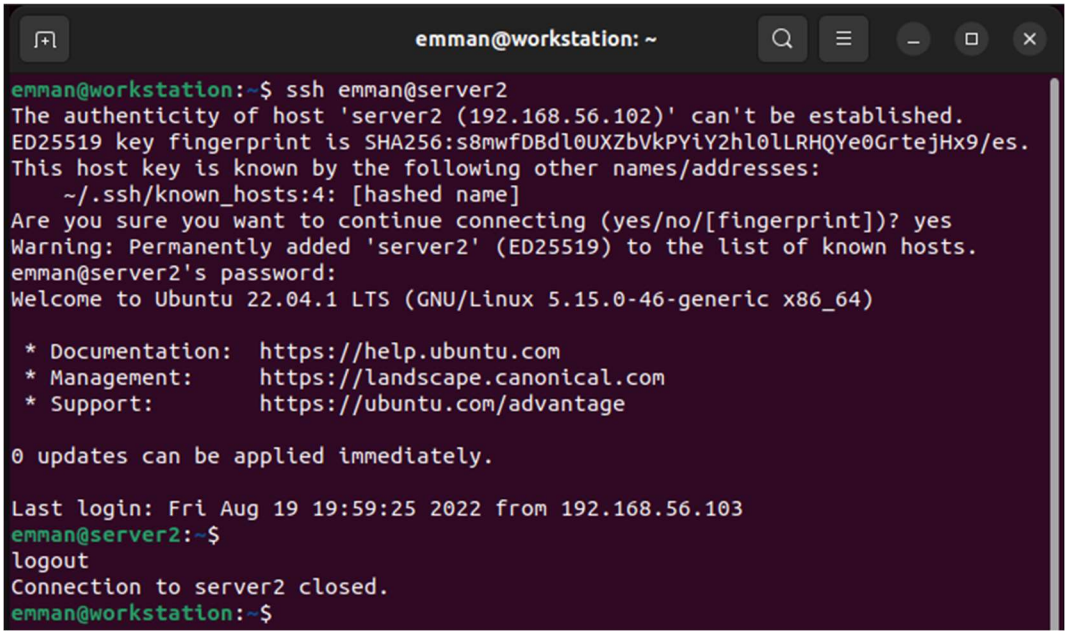


5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

Server 1:



Server 2:



Reflections:

Answer the following:

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

    - By completing step 4, in which we update the /etc/hosts file and add the IP addresses of servers 1 and 2, matching their names, we are allowed to use IP addresses in SSH commands. By doing this, we are creating an alias for an IP address that can be used to access SSH servers using just the server's login.

2. How secured is SSH?

    - By default, SSH connects to the remote machine via the TCP protocol on port 22, but by default, this connection is not secure. To send encrypted packets to a recipient system, SSH first encrypts the packets before receiving them, and vice versa. The use of a multiplex multiple connected by SSH, which builds a tunnel over the TCP connection, is the other factor keeping it linked. When connecting to a machine that offers a high level of security, it also employs authentication, but this can occasionally be brute forced.

Honor Pledge:

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