Name: BONIFACIO, REDJ GUILLIAN F.	**Date Performed:8/6/2025**
Course/Section: CPE3124	**Date Submitted:8/6/2025**
Instructor: ROBIN VALENZUELA	**Semester and SY: 1ST 2025**

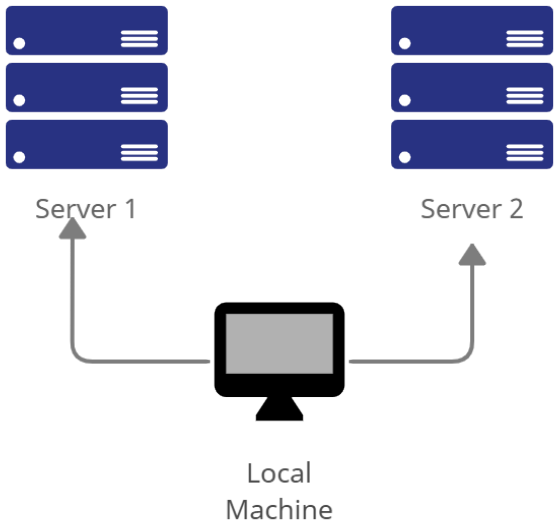
Activity 1: Configure Network using Virtual Machines

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:**

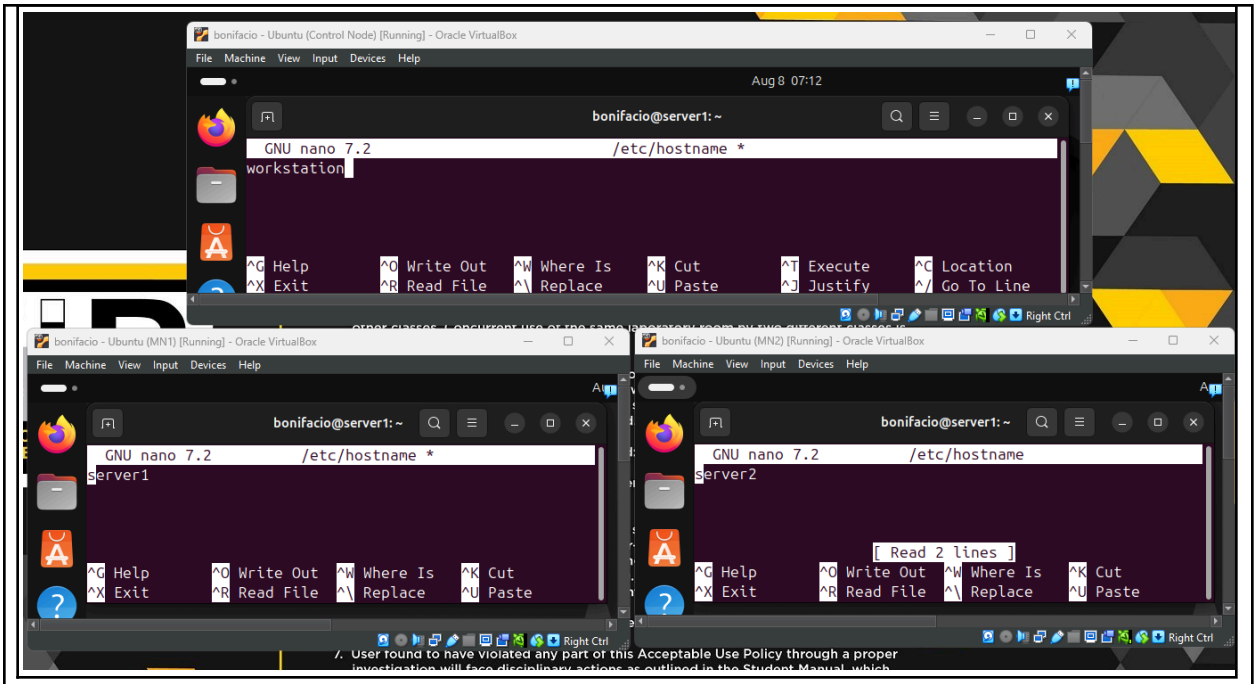
Network Topology:

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*).

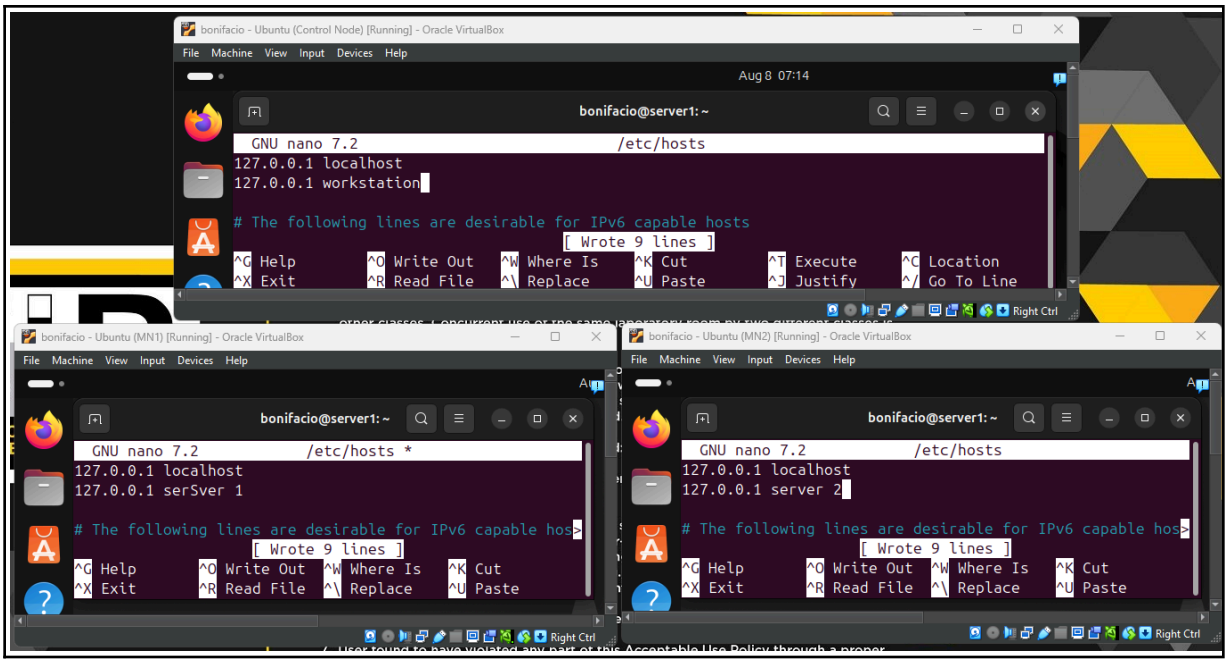


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

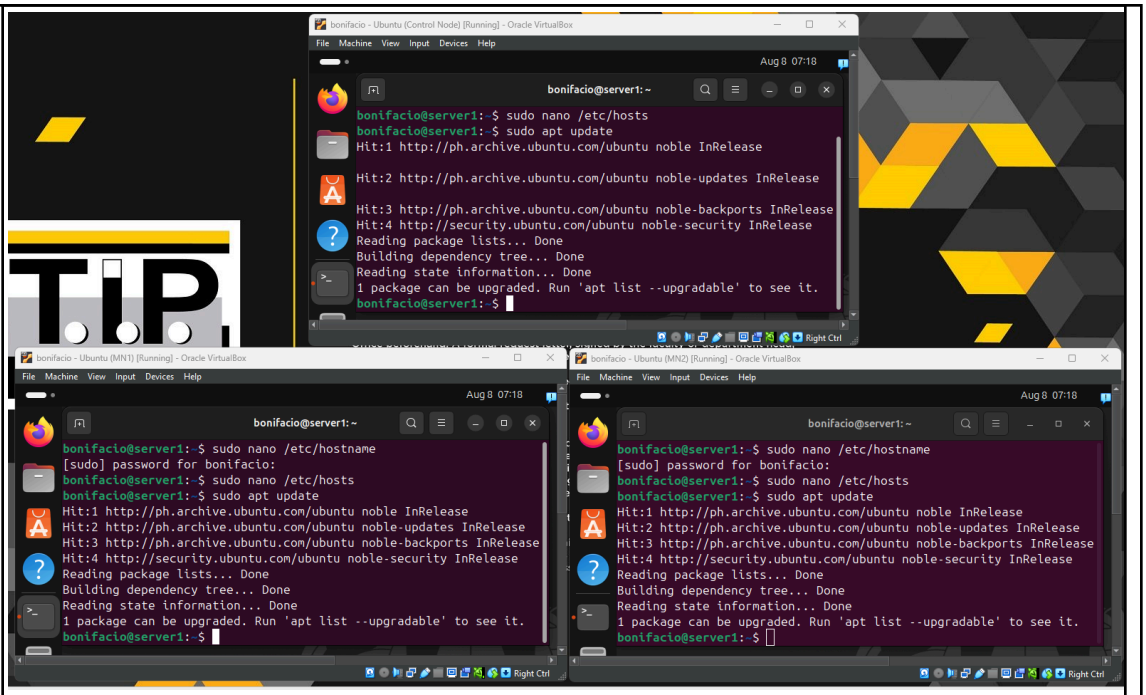


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1
   2.2 Type 127.0.0.1 server 2 for Server 2
   2.3 Type 127.0.0.1 workstation for the Local Machine

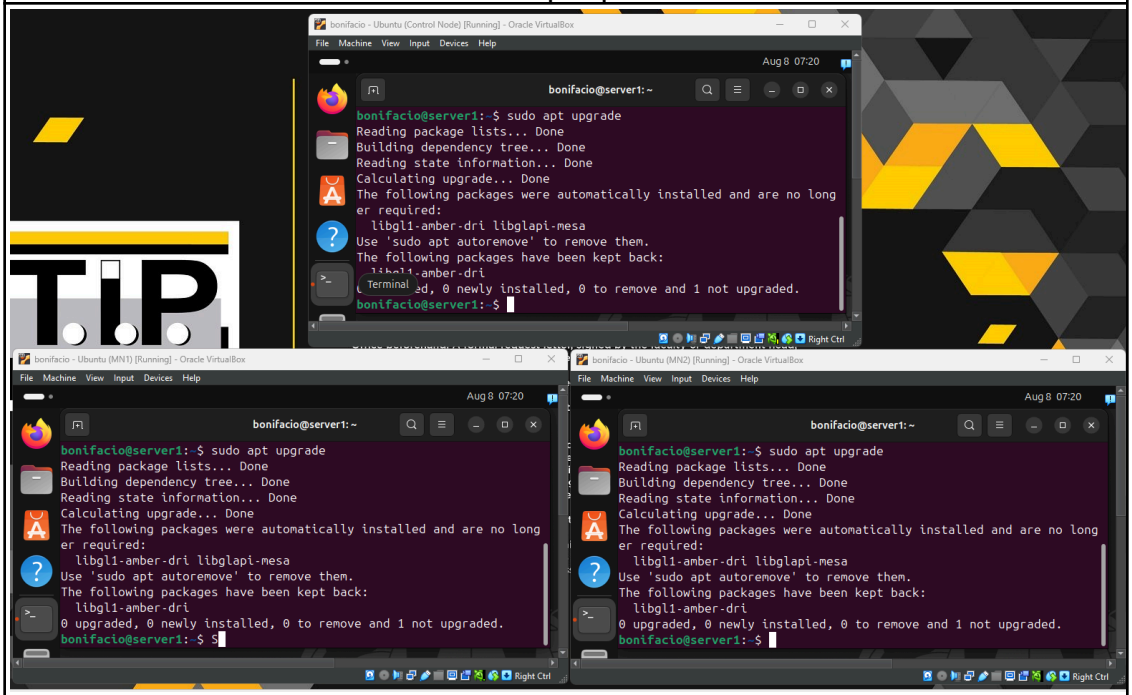


Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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

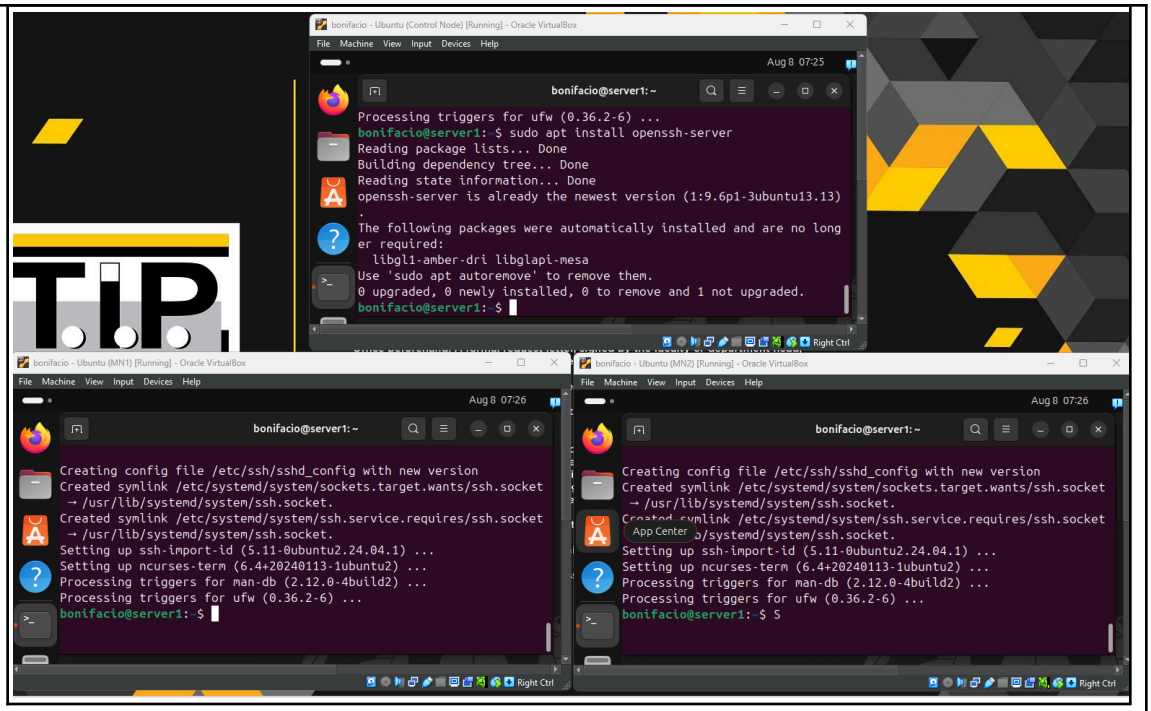


sudo apt update



sudo apt upgrade

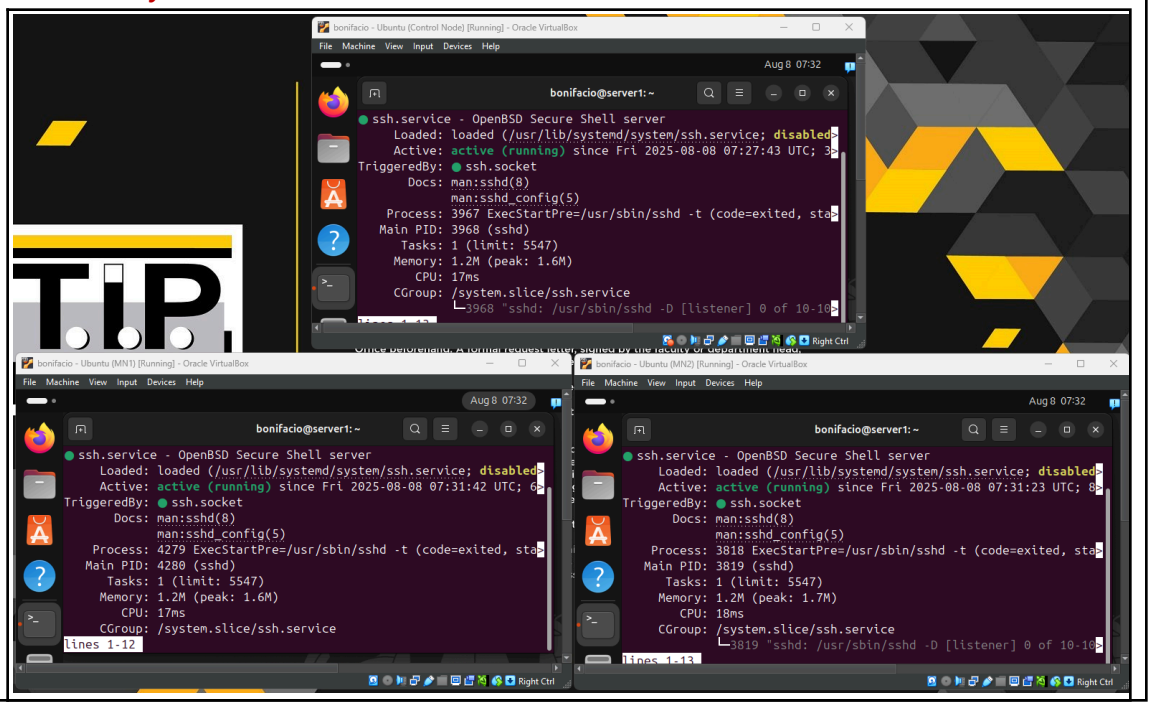
2. Install the SSH server using the command *sudo apt install openssh-server*.

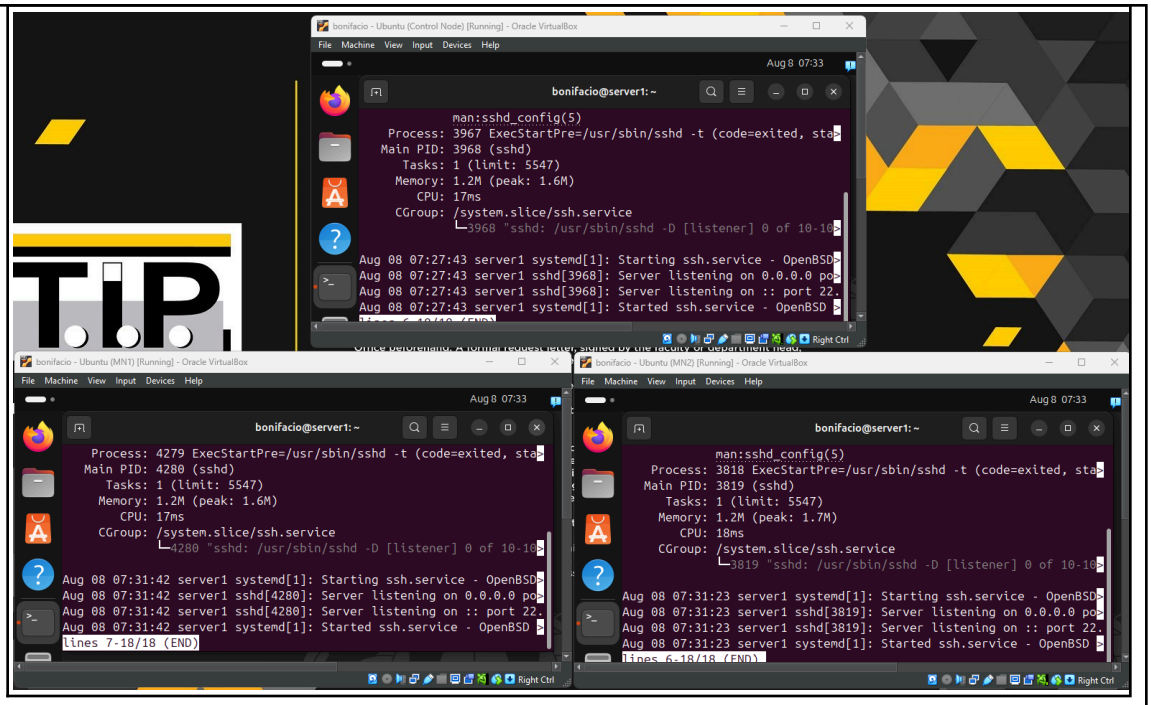


3. Verify if the SSH service has started by issuing the following commands:

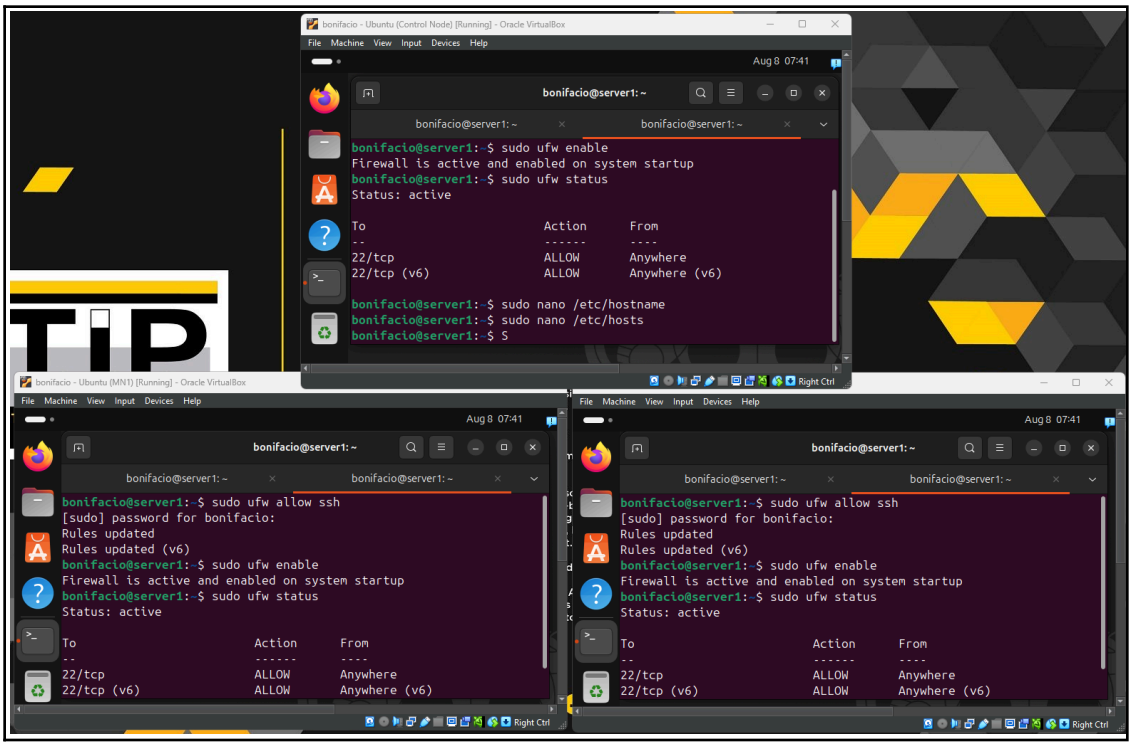
   3.1 sudo service ssh start

   3.2 sudo systemctl status ssh



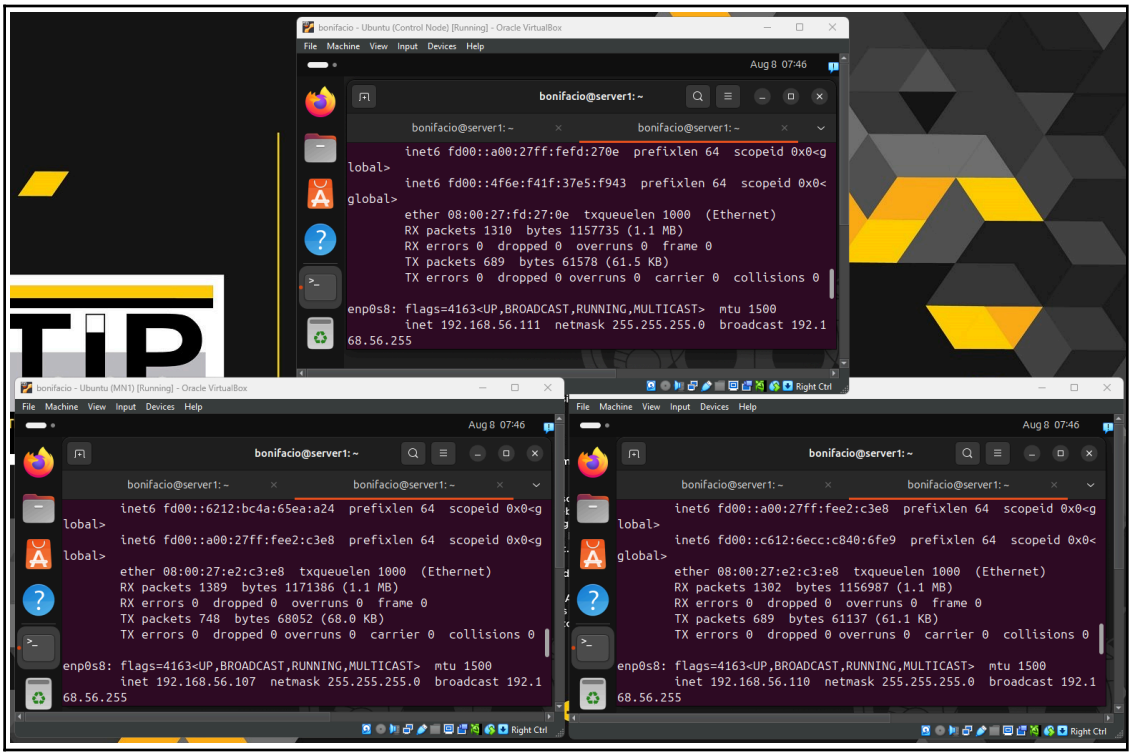


4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 sudo ufw allow ssh
   4.2 sudo ufw enable
   4.3 sudo ufw status

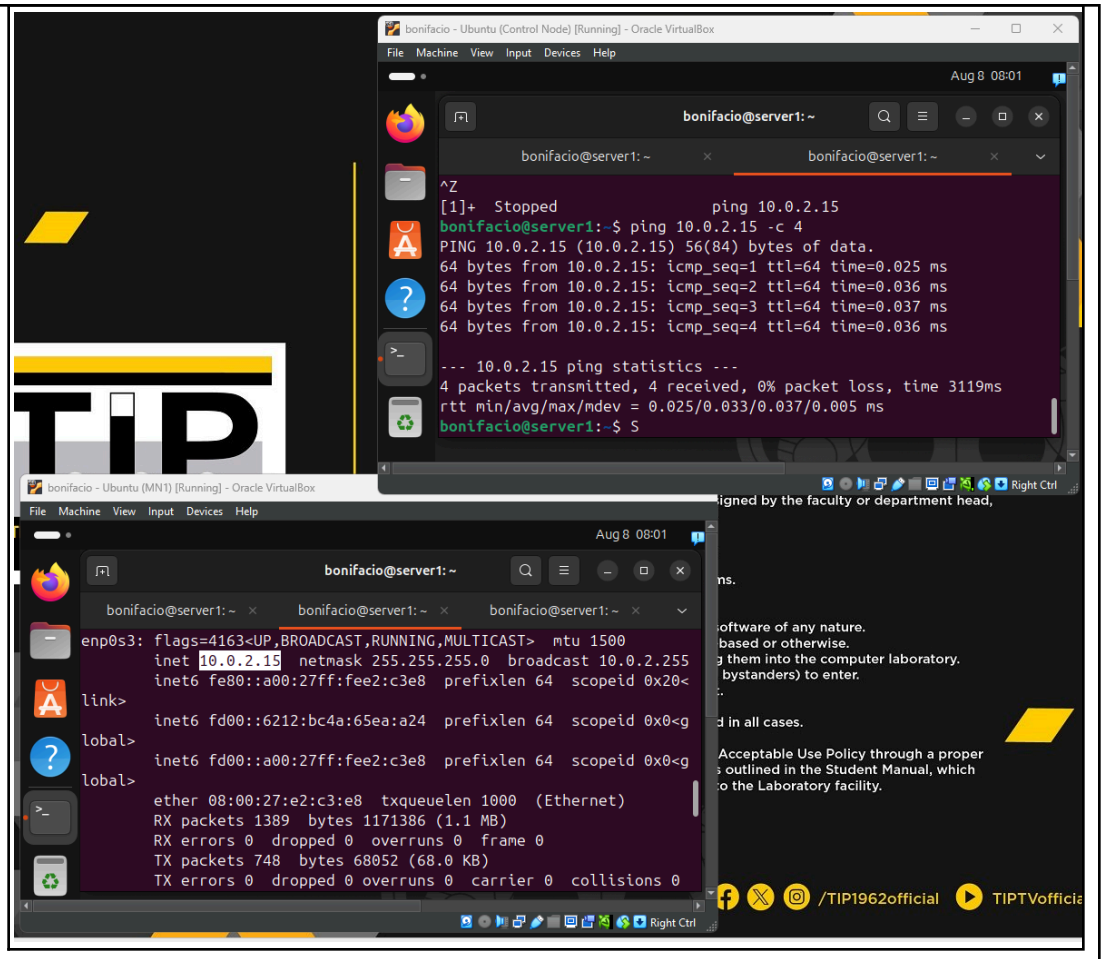


Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

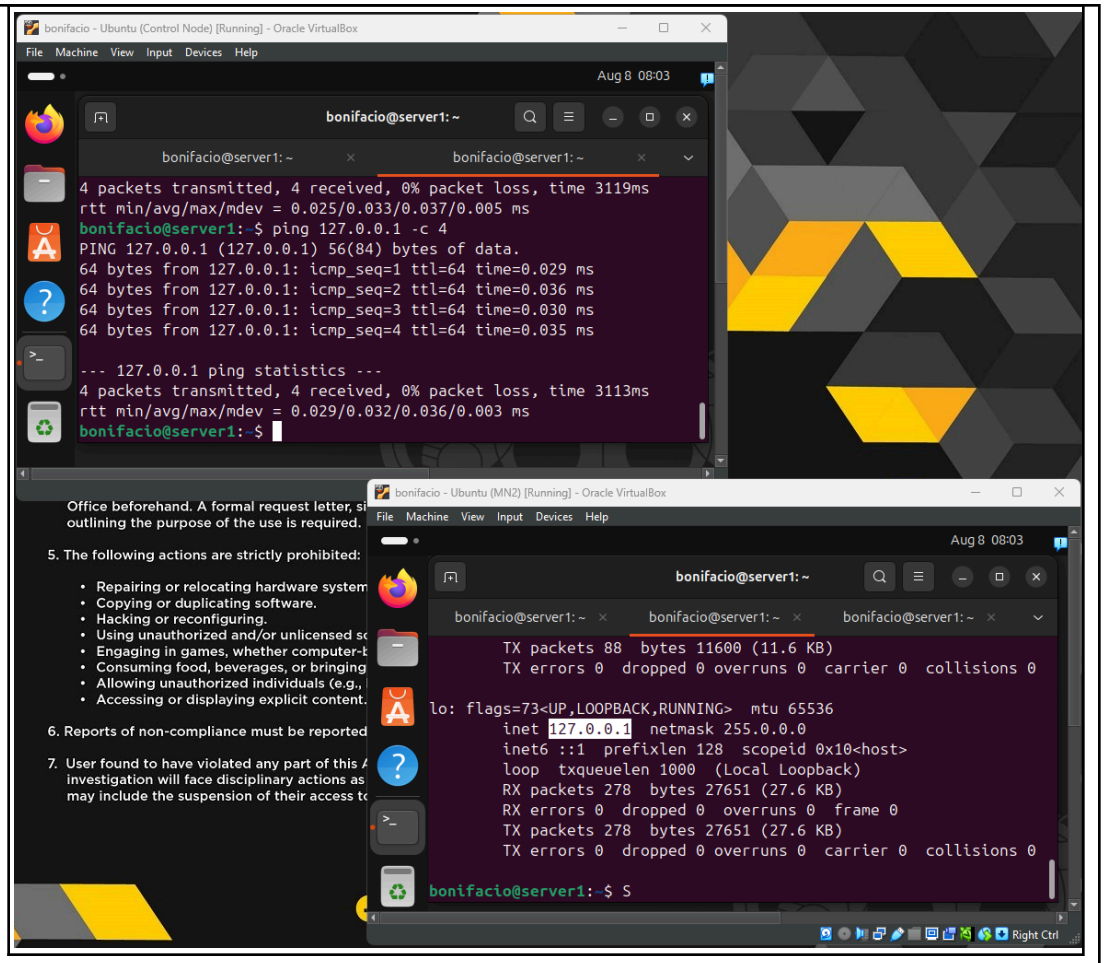
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.
   1.1 Server 1 IP address: 192.168.56.107
   1.2 Server 2 IP address: 192.168.56.110
   1.3 Server 3 IP address: 192.168.56.111



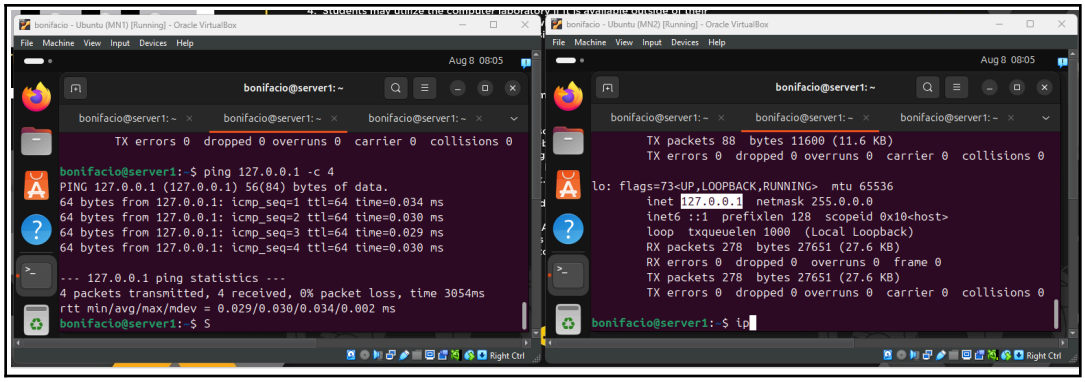
2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ☐ Successful ☐ Not Successful



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

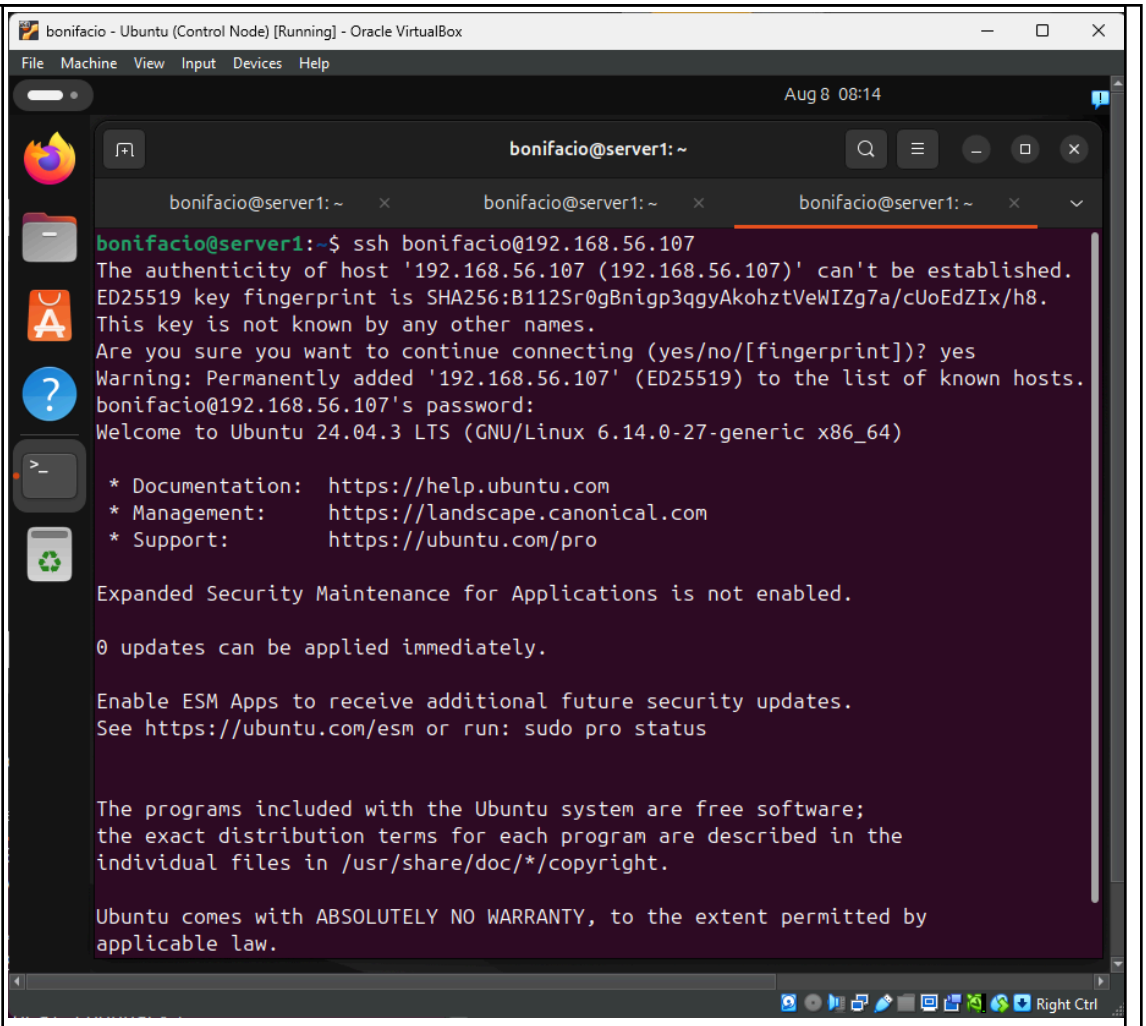


Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

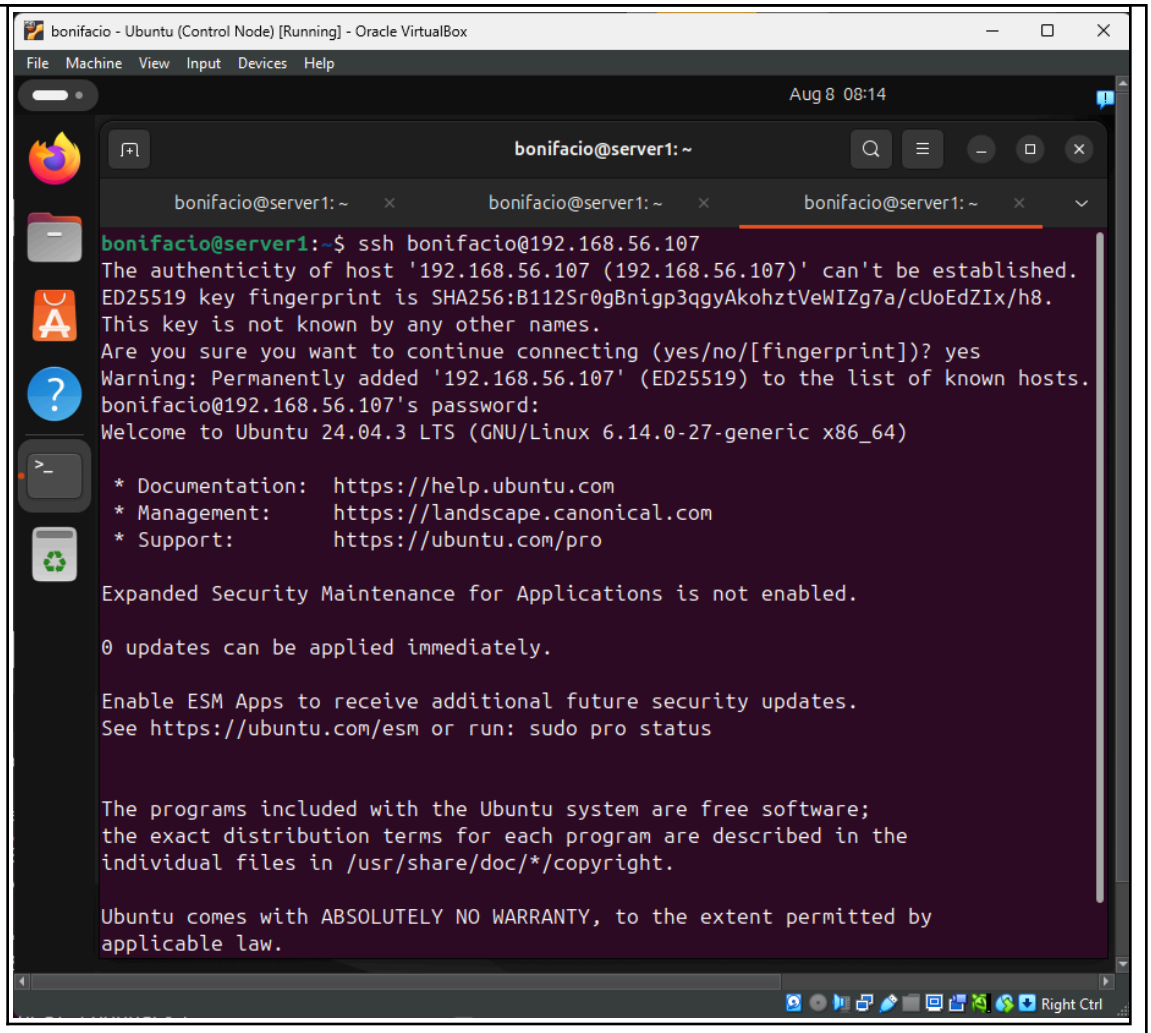
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted

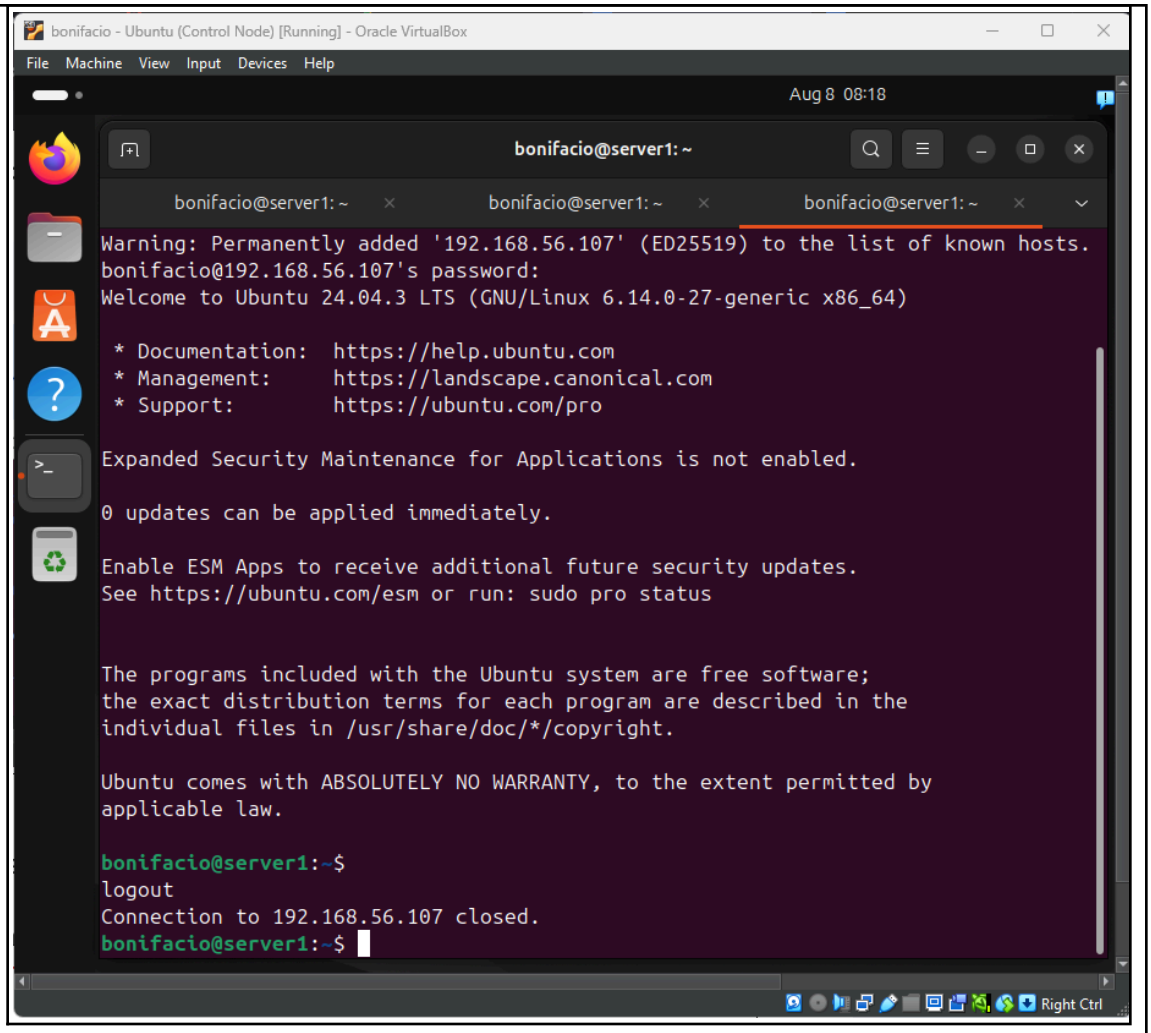


local machine

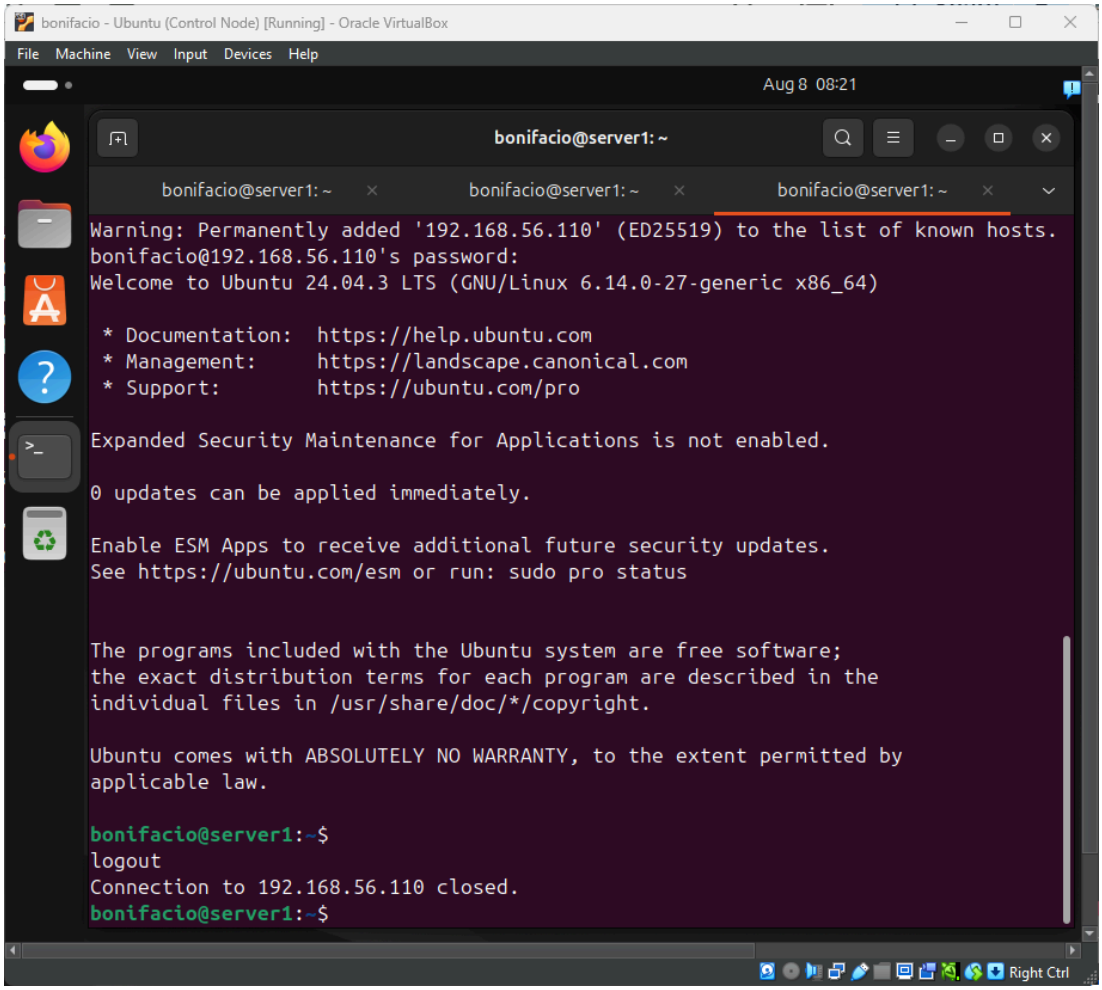
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*



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



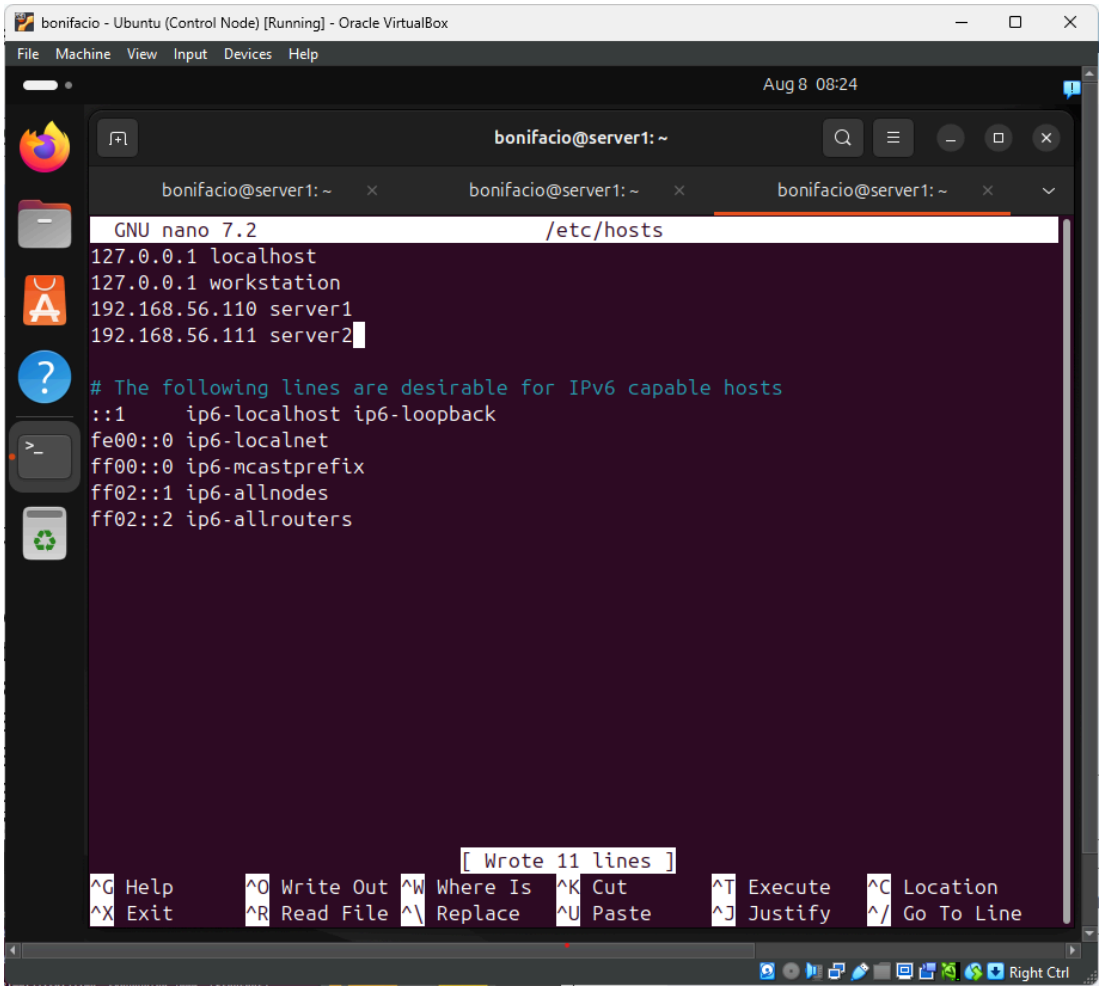
3. Do the same for Server 2.



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

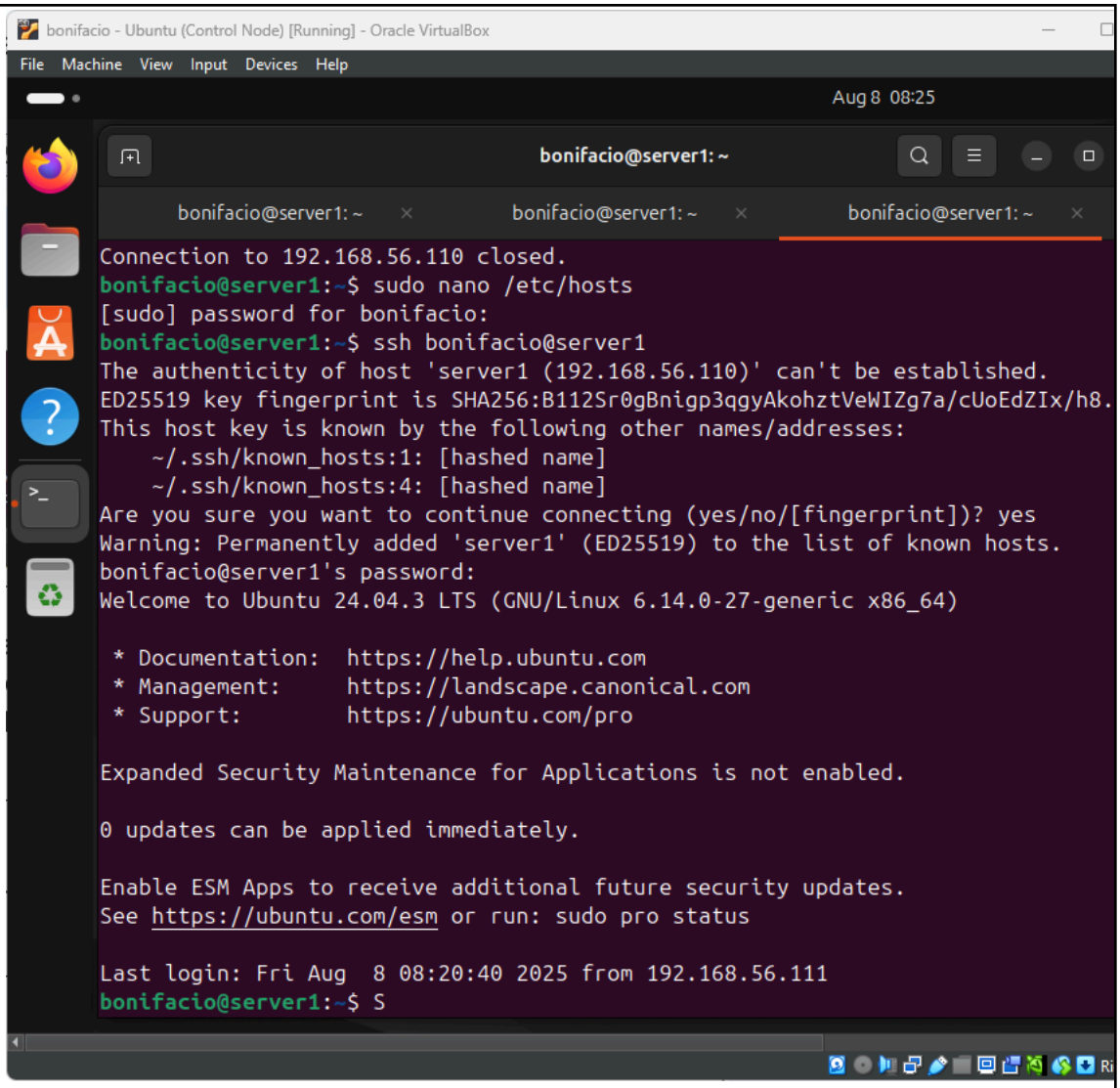
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

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.



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   In SSH commands, you can use the hostname instead of an IP address because the SSH client depends on the operating system's name resolution system to translate the hostname into its corresponding IP address. This resolution is typically handled through DNS servers, the local `/etc/hosts` file, or other network name resolution services. When you enter `ssh user@hostname`, your system first looks up the IP address associated with the hostname using these methods before establishing a connection to the SSH daemon on the remote machine.

2. How secured is SSH?

When configured correctly, SSH is considered highly secure. It encrypts authentication credentials, preventing eavesdropping and protecting against man-in-the-middle attacks by securing all communication between the client and server. SSH supports advanced authentication methods like public-key authentication and multi-factor authentication. With its strong encryption algorithms and secure key exchange processes, it ensures both data confidentiality and integrity. The security of SSH depends on best practices such as disabling root login, using strong passwords or keys, keeping the server updated, and managing user access carefully. When implemented properly, SSH is a reliable and trusted solution for secure remote services and system administration.