Name: Kenneth Gabriel A. Llave	**Date Performed: 14/08/2023**
Course/Section: CPE31S4	**Date Submitted: 15/08/2023**
Instructor: Jonathan Taylar	**Semester and SY: 1st sem 2023-2024**

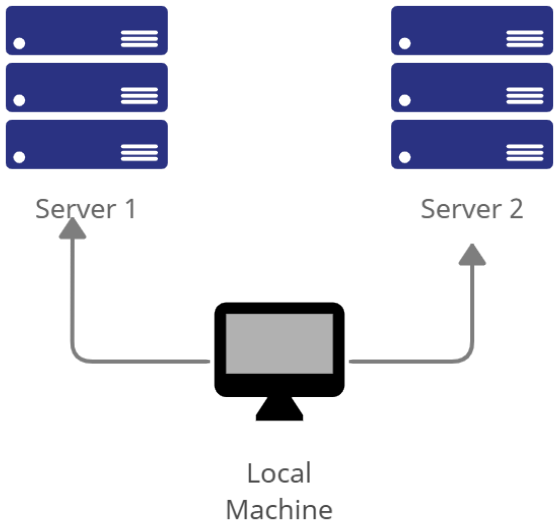
Activity 1: Configure Network using Virtual Machines

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:

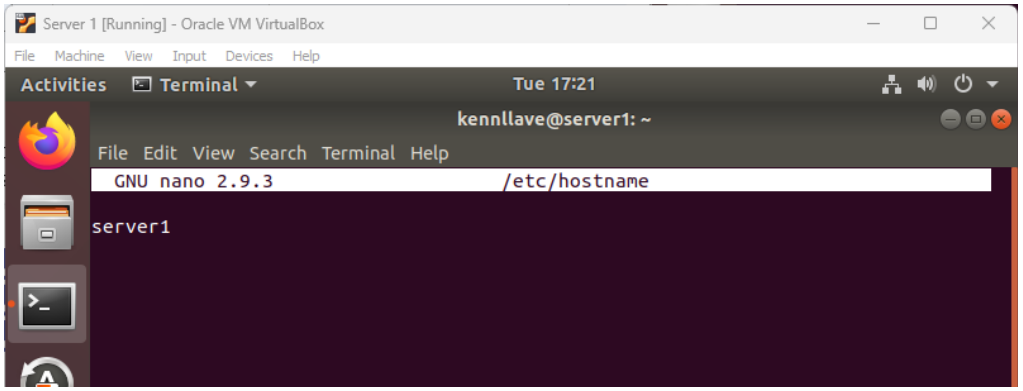
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



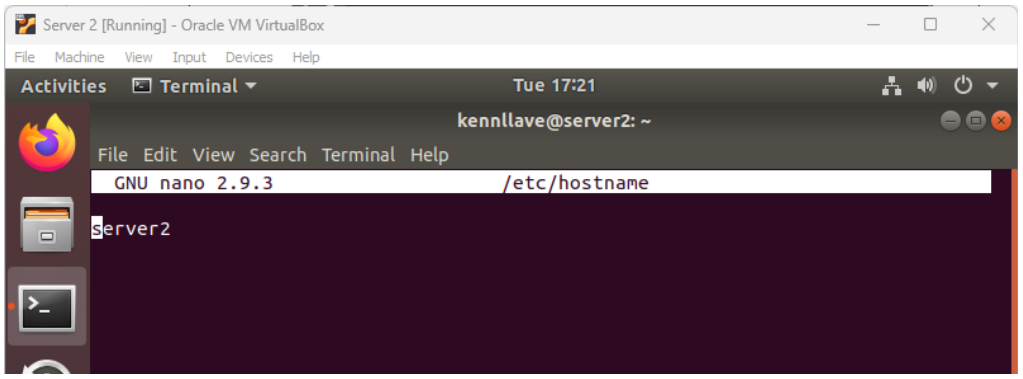
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

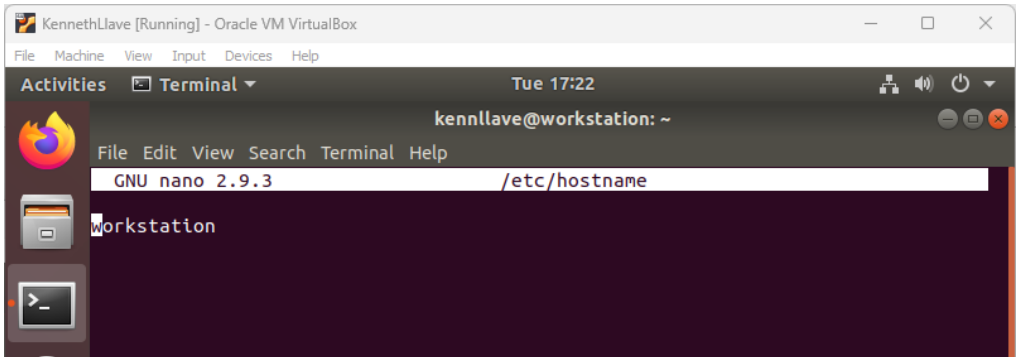
1.1 Use server1 for Server 1



1.2 Use server2 for Server 2

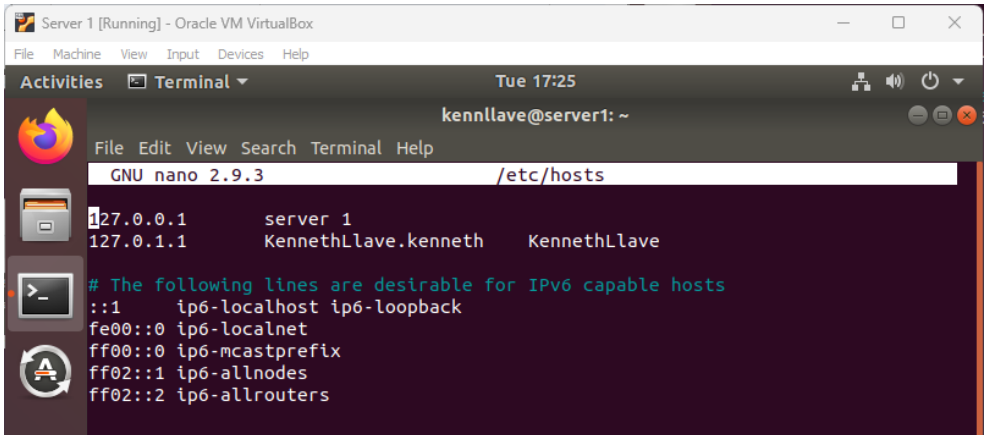


1.3 Use workstation for the Local Machine

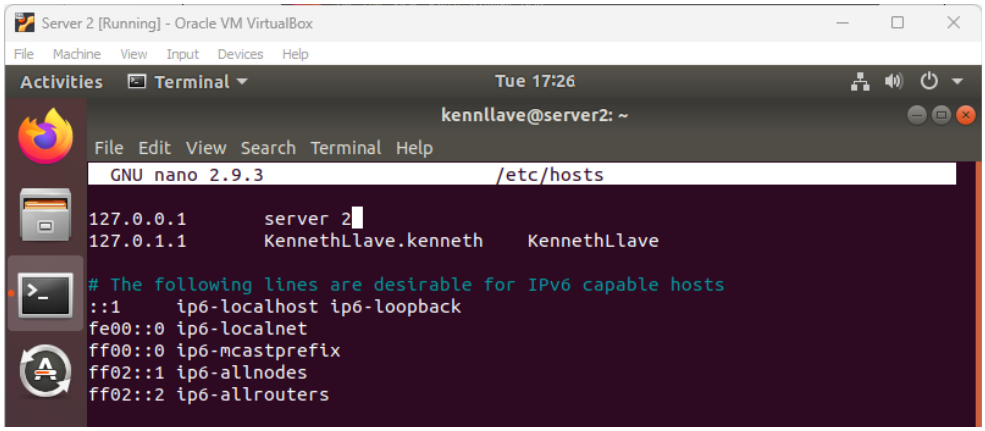


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

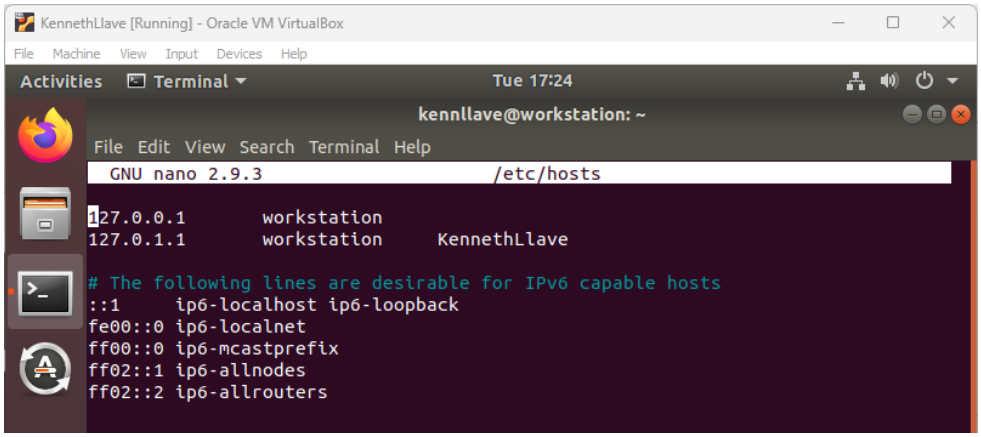
2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2

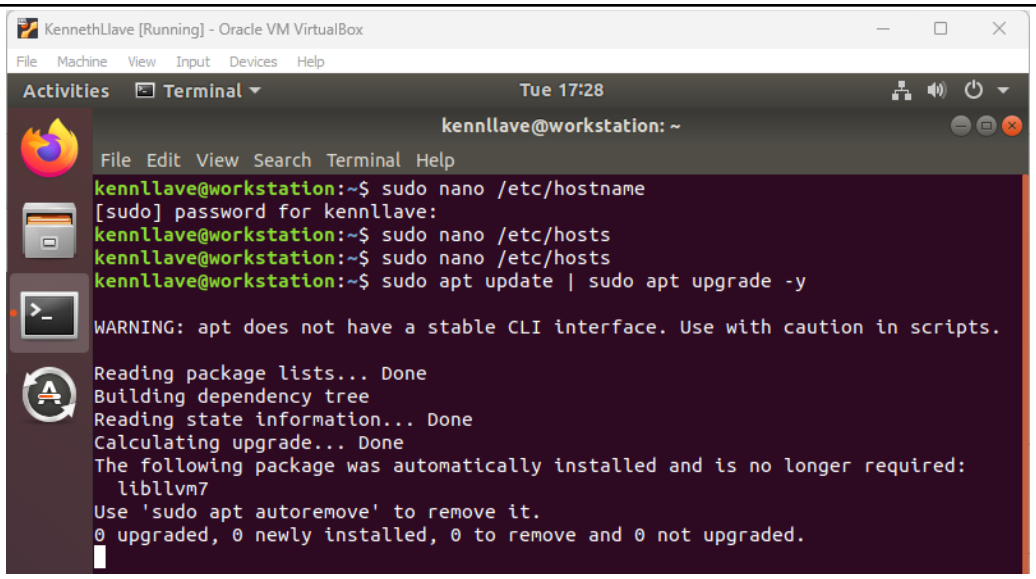


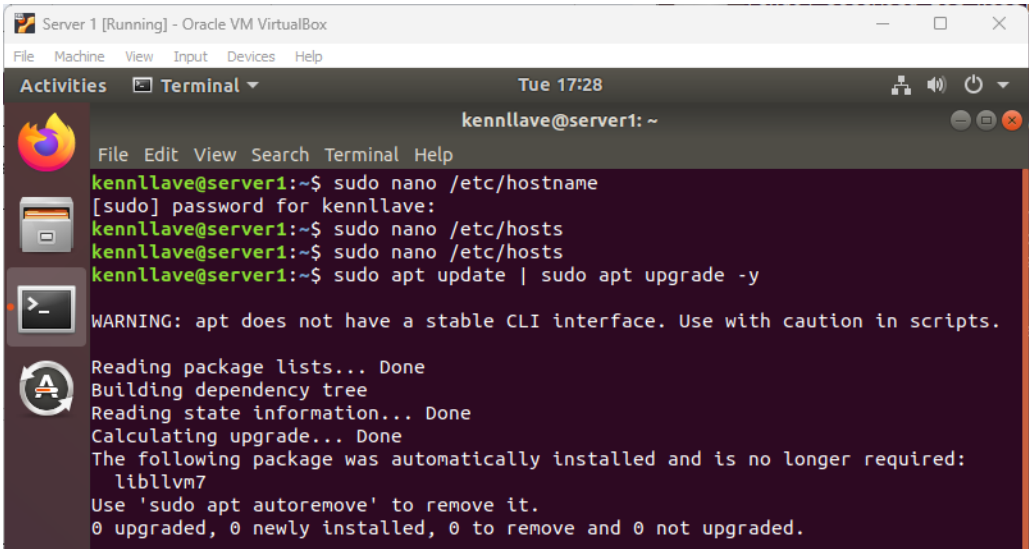
2.3 Type 127.0.0.1 workstation for the Local Machine

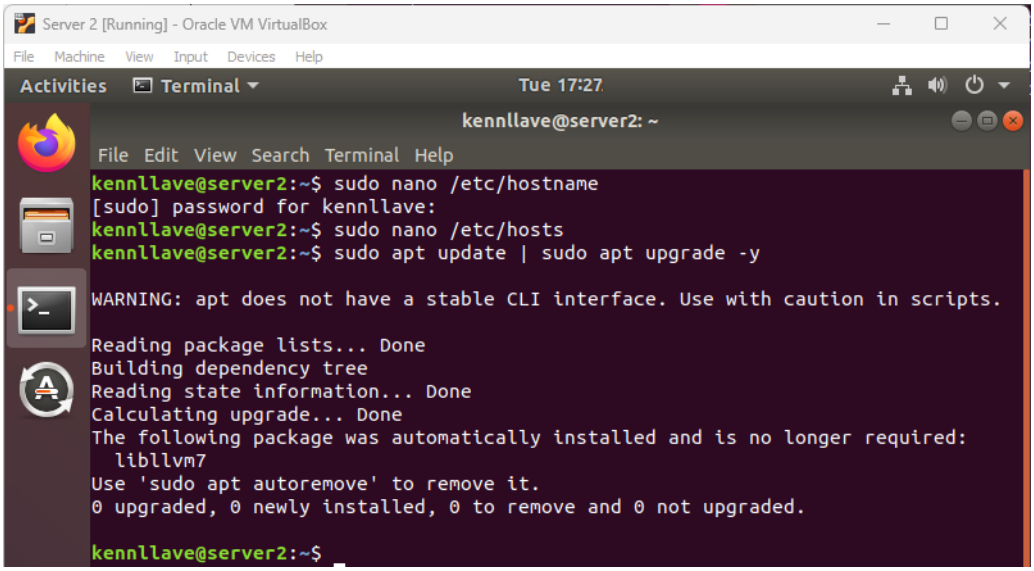


Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

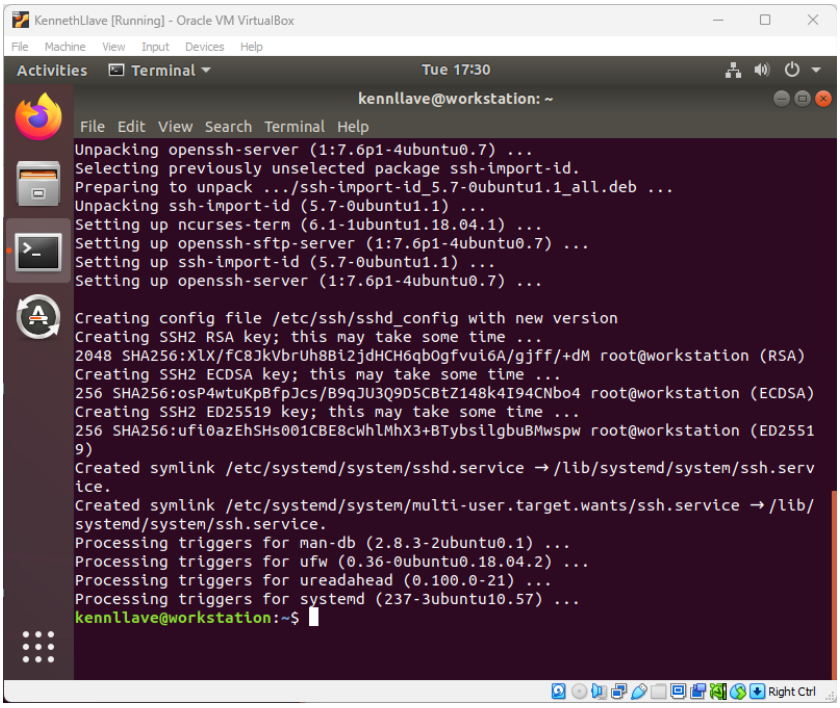
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.

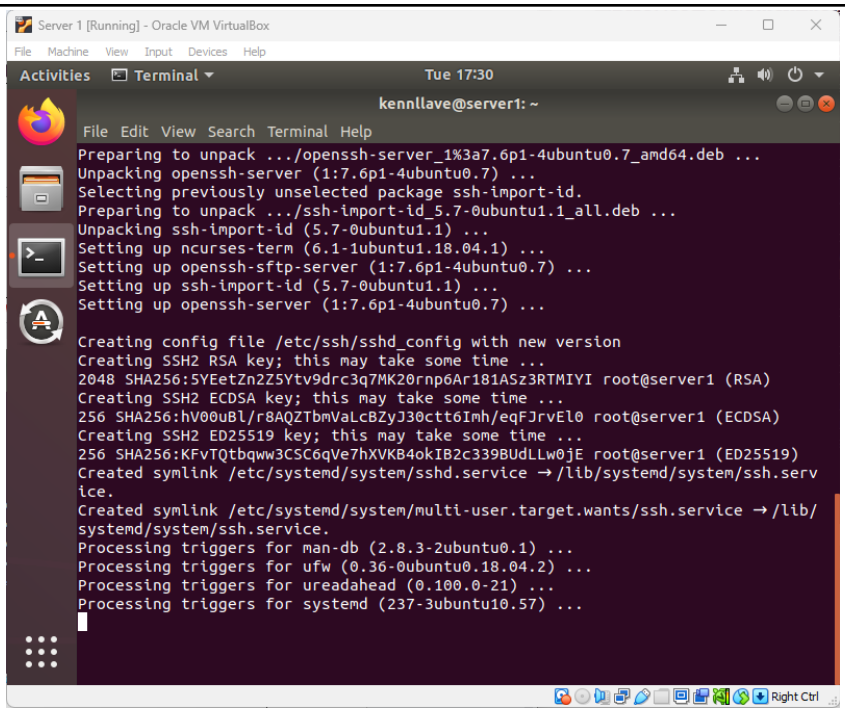

KennethLlave [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Tue 17:28
kennllave@workstation: ~
File Edit View Search Terminal Help
kennllave@workstation:~$ sudo nano /etc/hostname
[sudo] password for kennllave:
kennllave@workstation:~$ sudo nano /etc/hosts
kennllave@workstation:~$ sudo nano /etc/hosts
kennllave@workstation:~$ sudo apt update | sudo apt upgrade -y
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

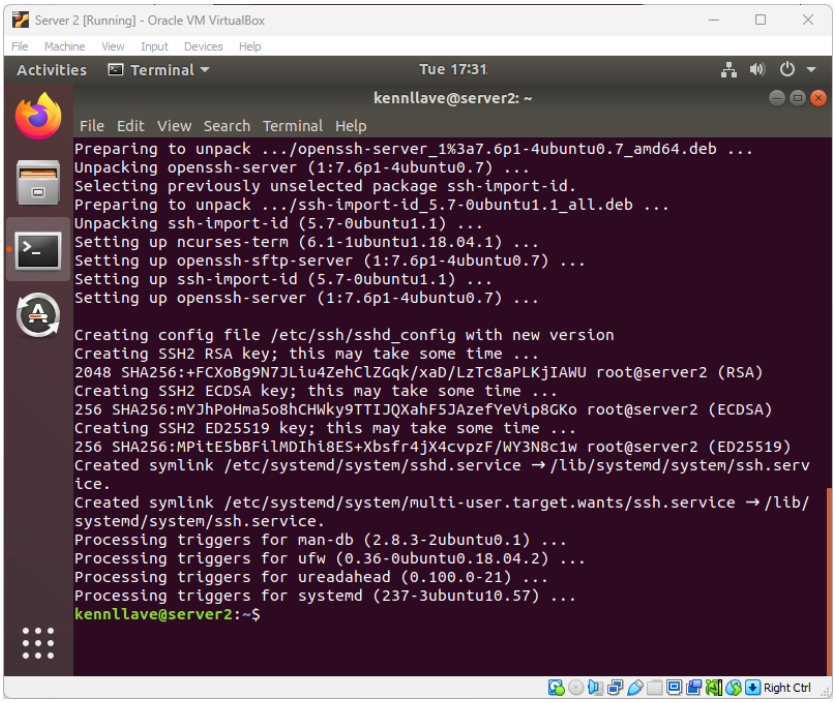

Server 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Tue 17:28
kennllave@server1: ~
File Edit View Search Terminal Help
kennllave@server1:~$ sudo nano /etc/hostname
[sudo] password for kennllave:
kennllave@server1:~$ sudo nano /etc/hosts
kennllave@server1:~$ sudo nano /etc/hosts
kennllave@server1:~$ sudo apt update | sudo apt upgrade -y
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.


Server 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Tue 17:27
kennllave@server2: ~
File Edit View Search Terminal Help
kennllave@server2:~$ sudo nano /etc/hostname
[sudo] password for kennllave:
kennllave@server2:~$ sudo nano /etc/hosts
kennllave@server2:~$ sudo apt update | sudo apt upgrade -y
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
kennllave@server2:~$

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

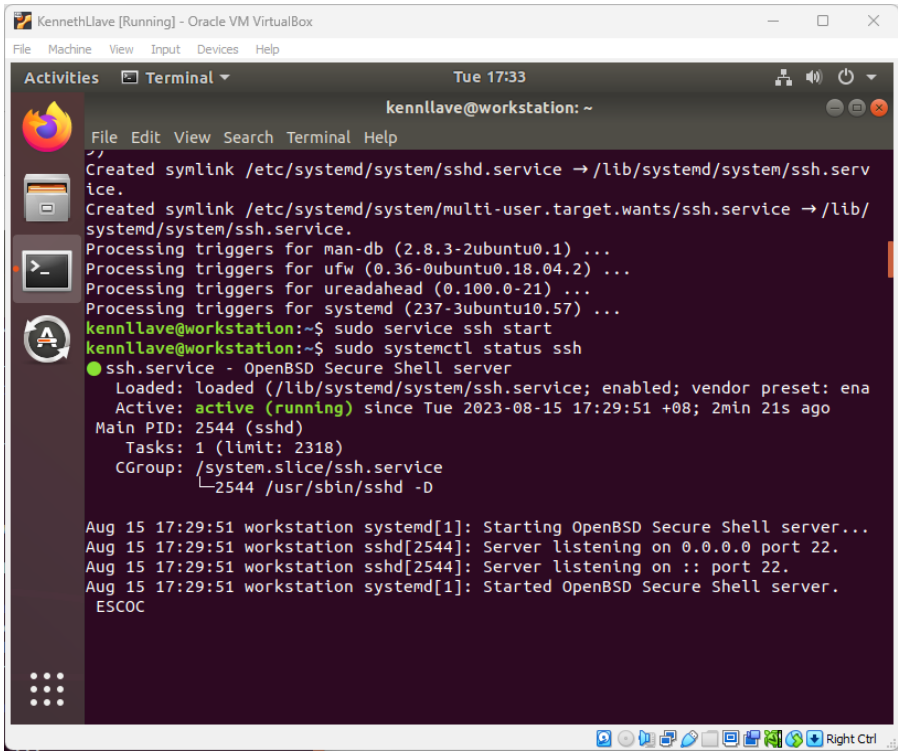


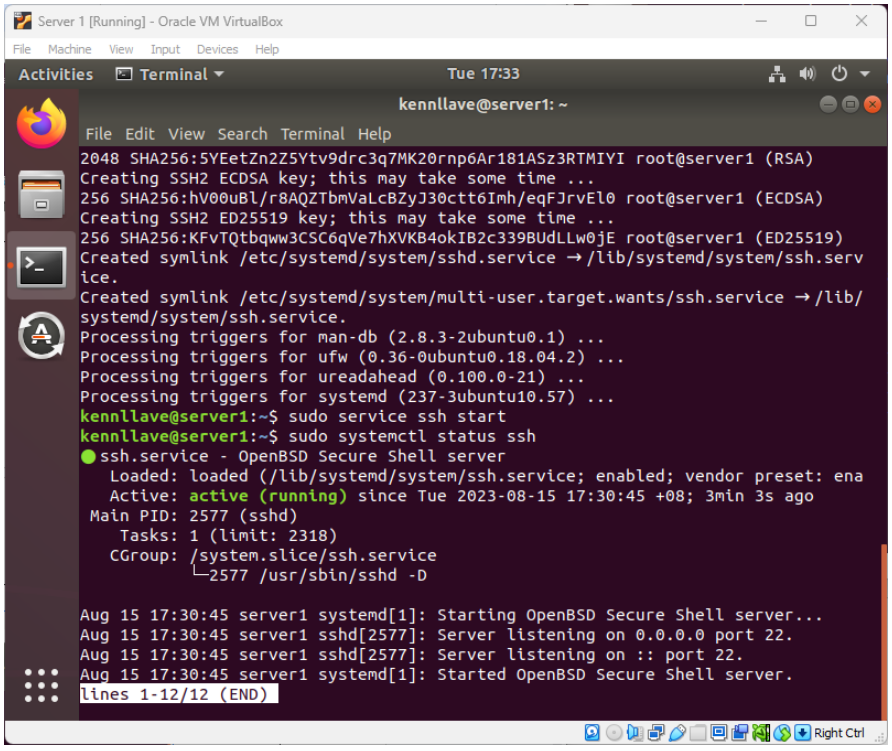


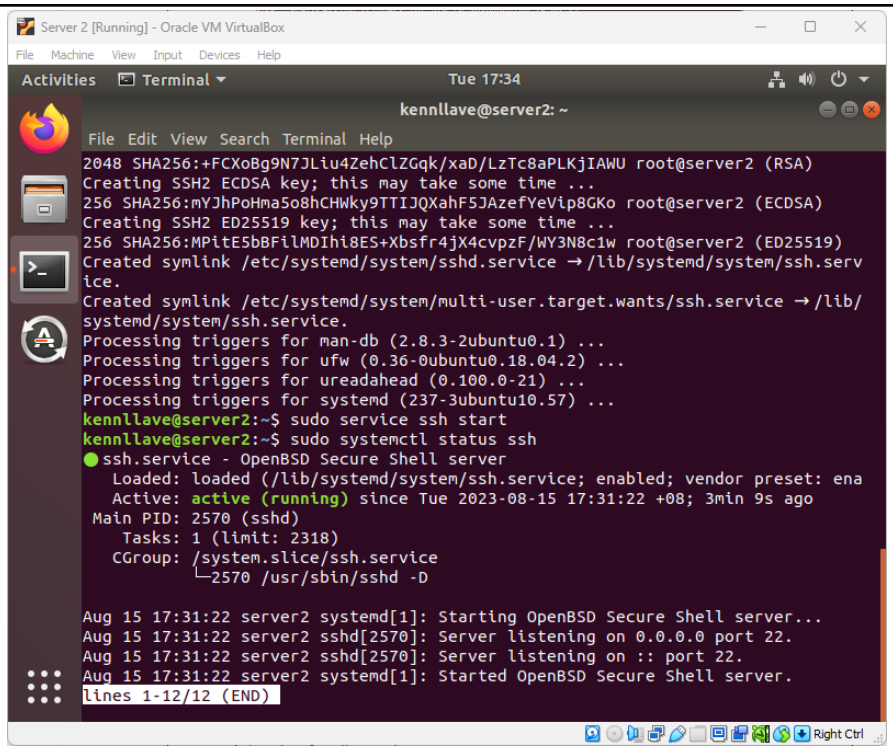


3. Verify if the SSH service has started by issuing the following commands:
   3.1 sudo service ssh start

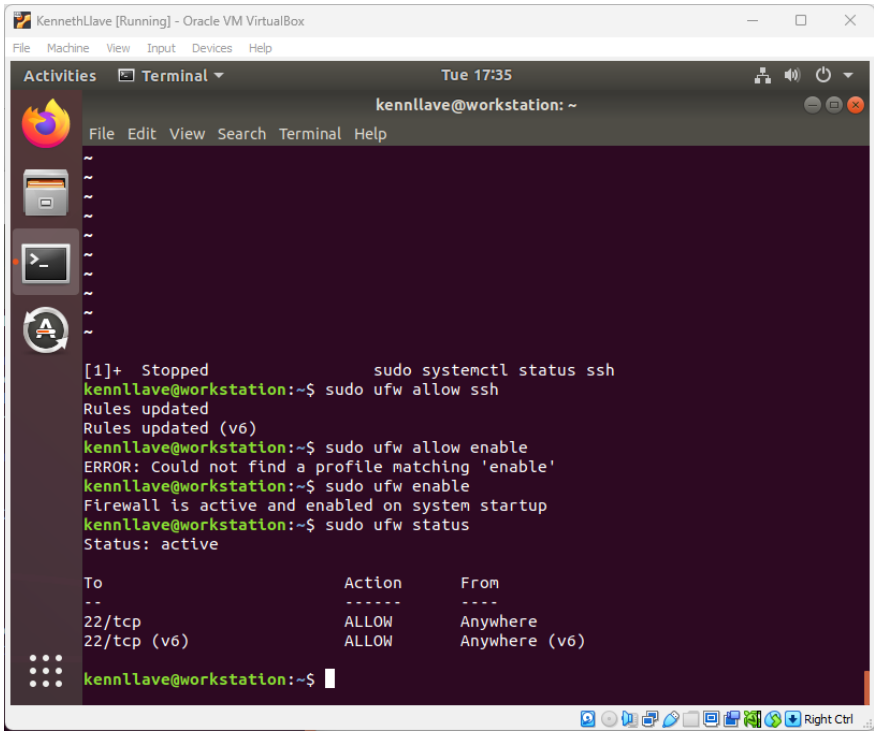
3.2 sudo systemctl status ssh

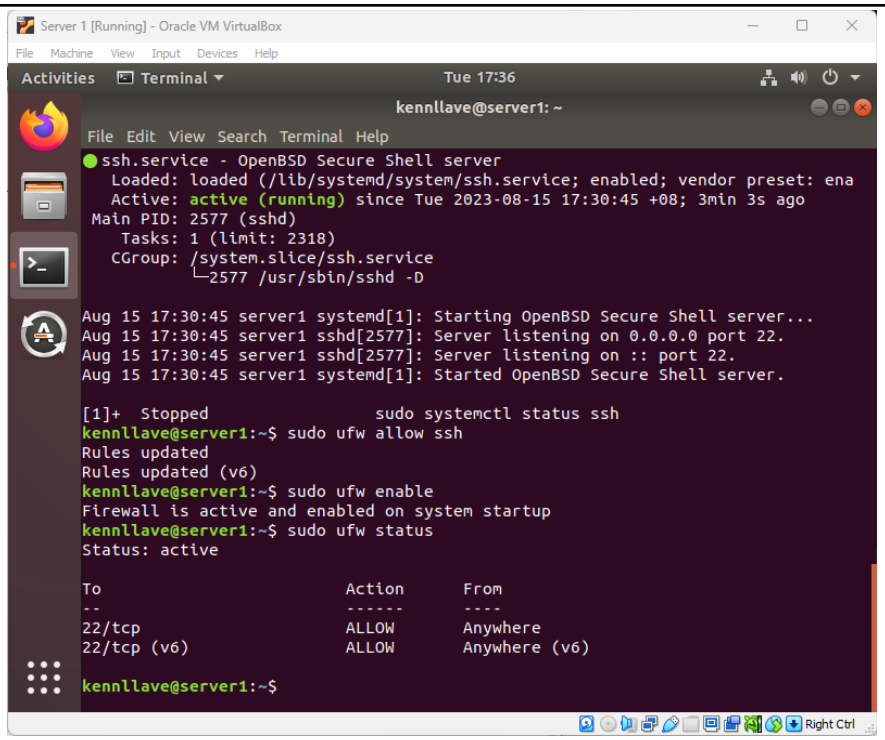


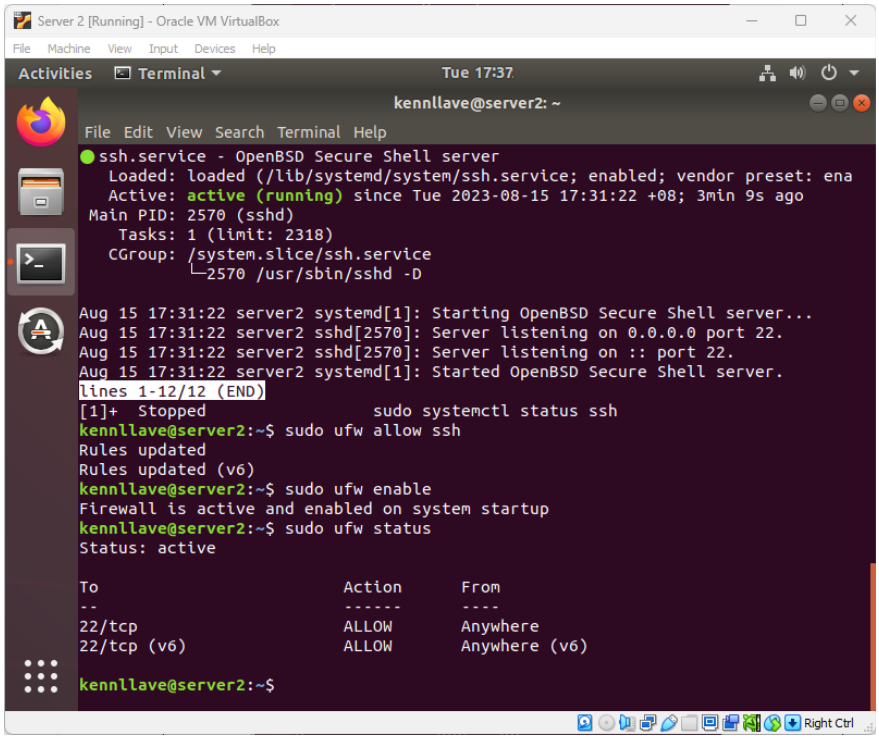




4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 sudo ufw allow ssh
   4.2 sudo ufw enable
   4.3 sudo ufw status





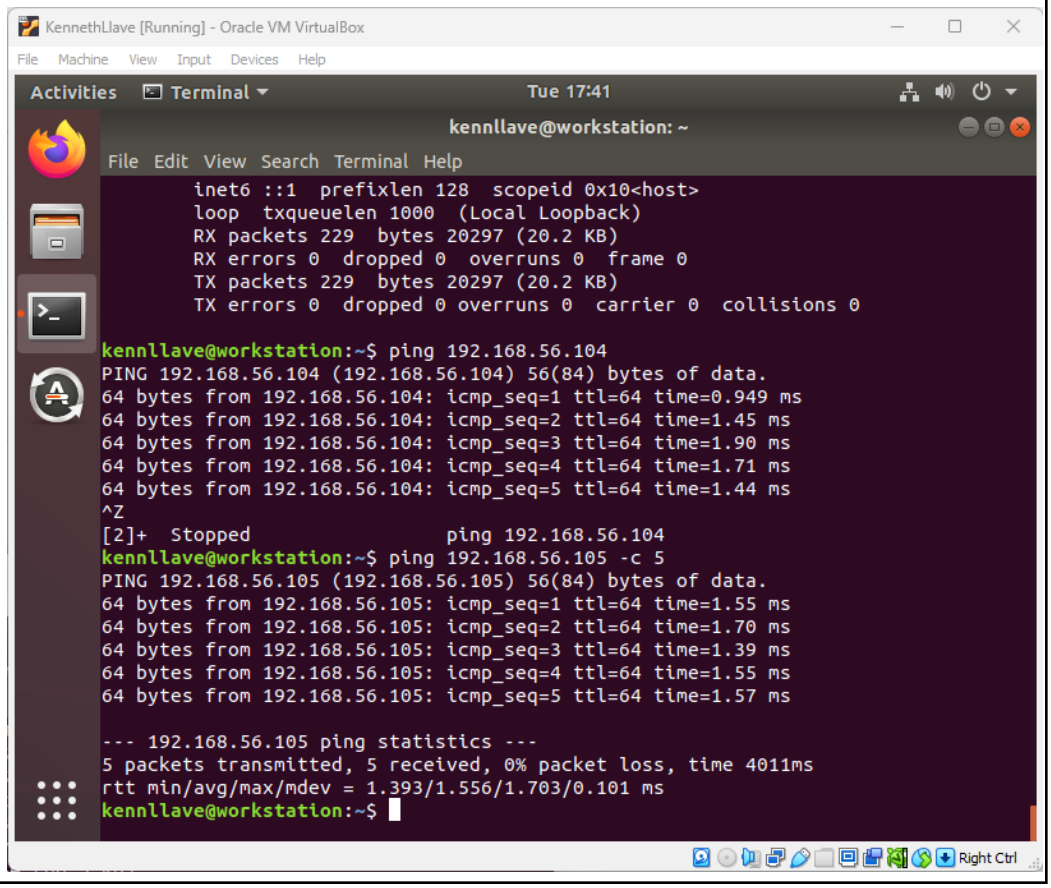


Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

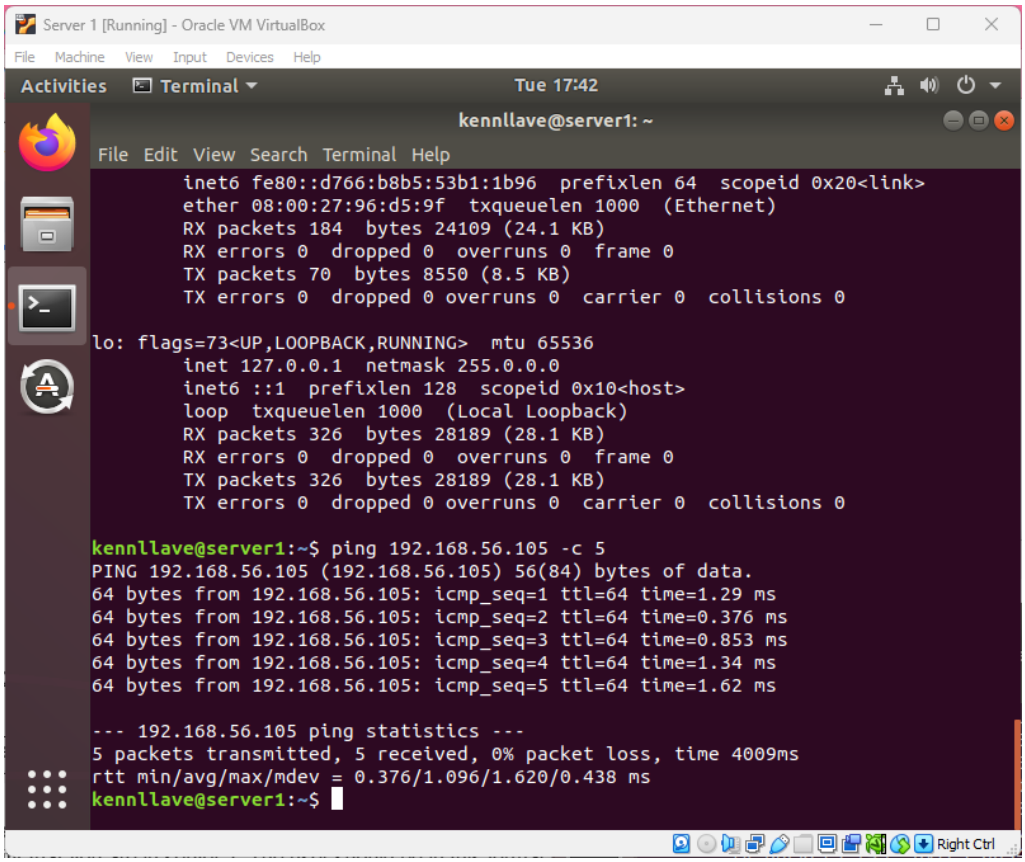
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.

1.1 Server 1 IP address: 192.168.56.104
1.2 Server 2 IP address: 192.168.56.105
1.3 Workstation IP address: 192.168.56.103

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

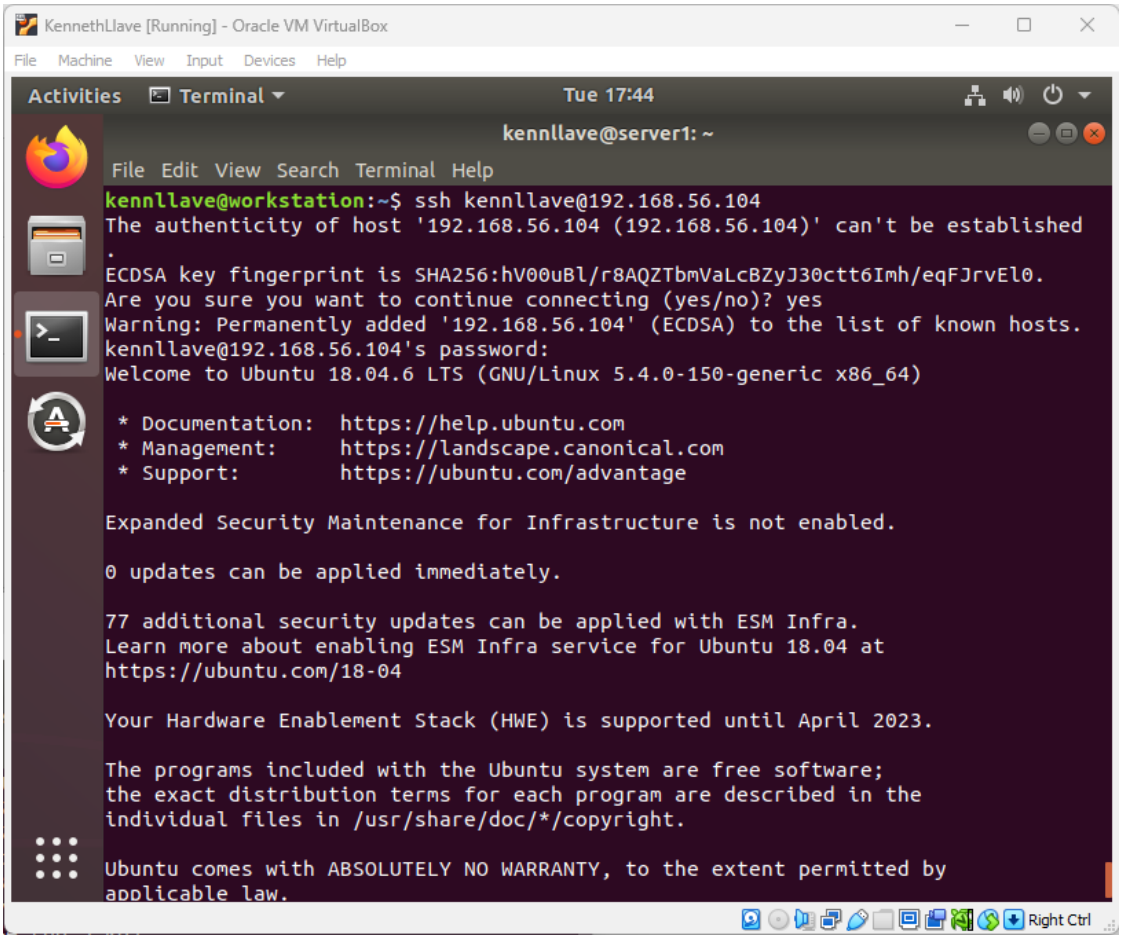


Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

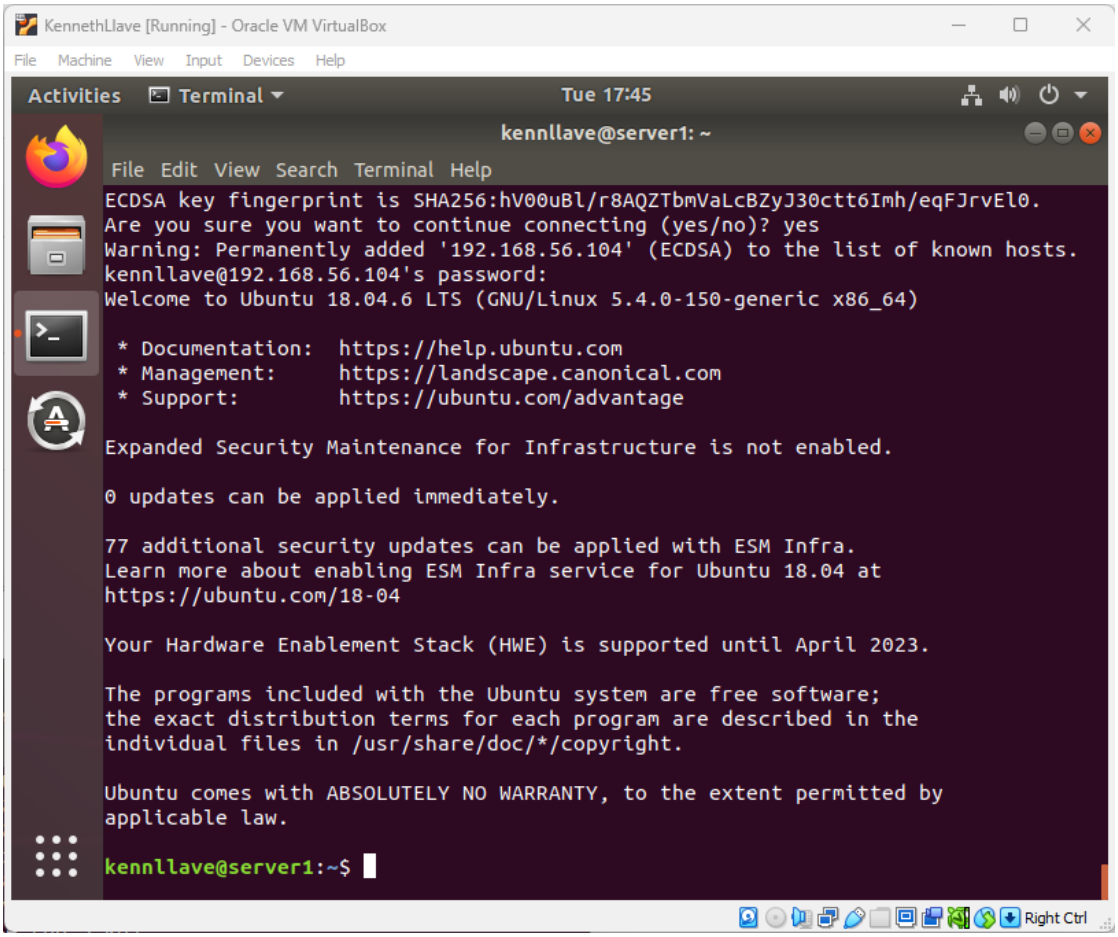
1. On the Local Machine, issue the following commands:

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

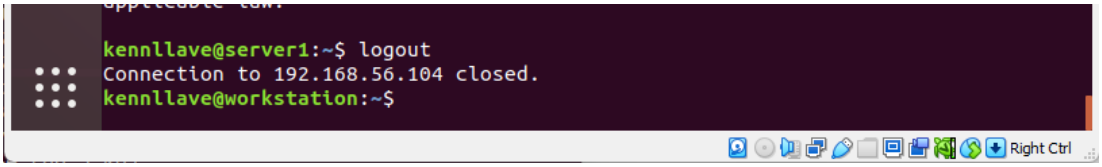
1.2 Enter the password for server 1 when prompted



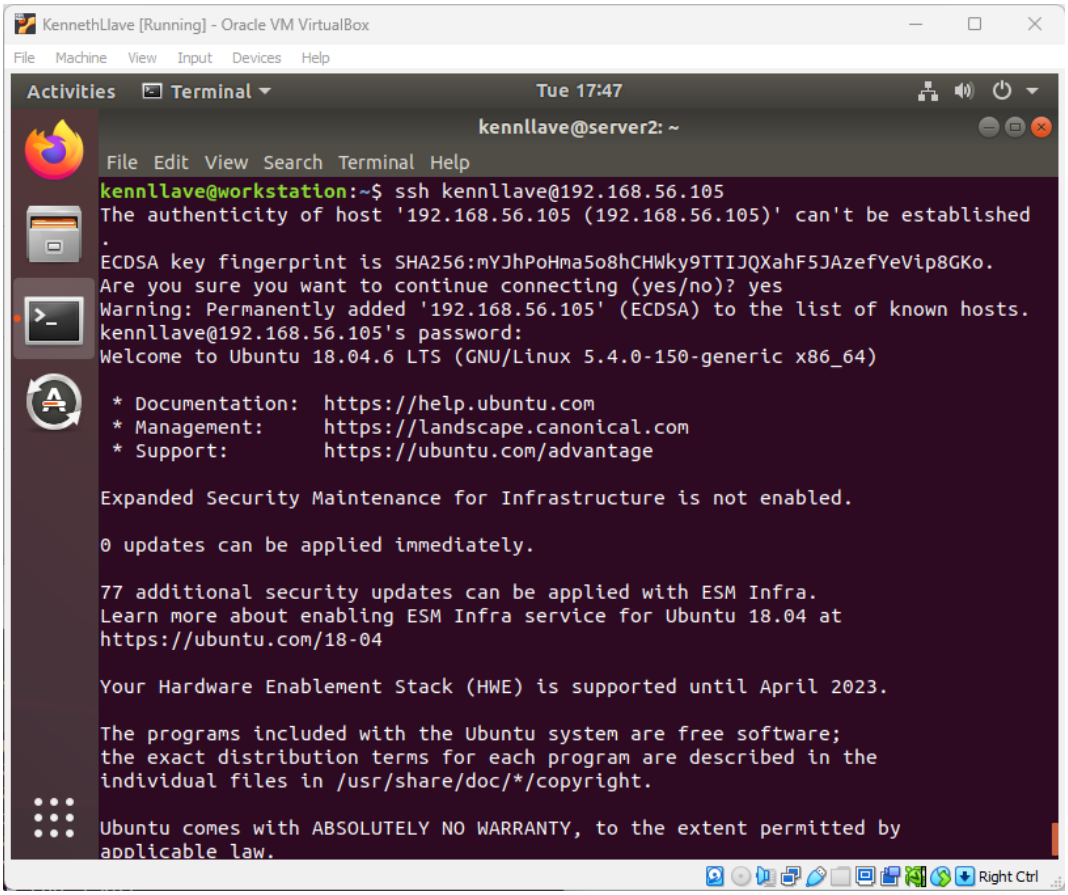
1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

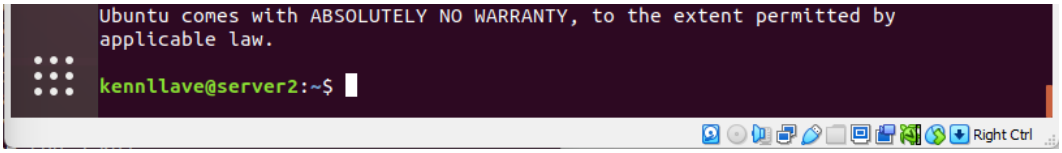


2. Logout of Server 1 by issuing the command *control + D.*



3. Do the same for Server 2.

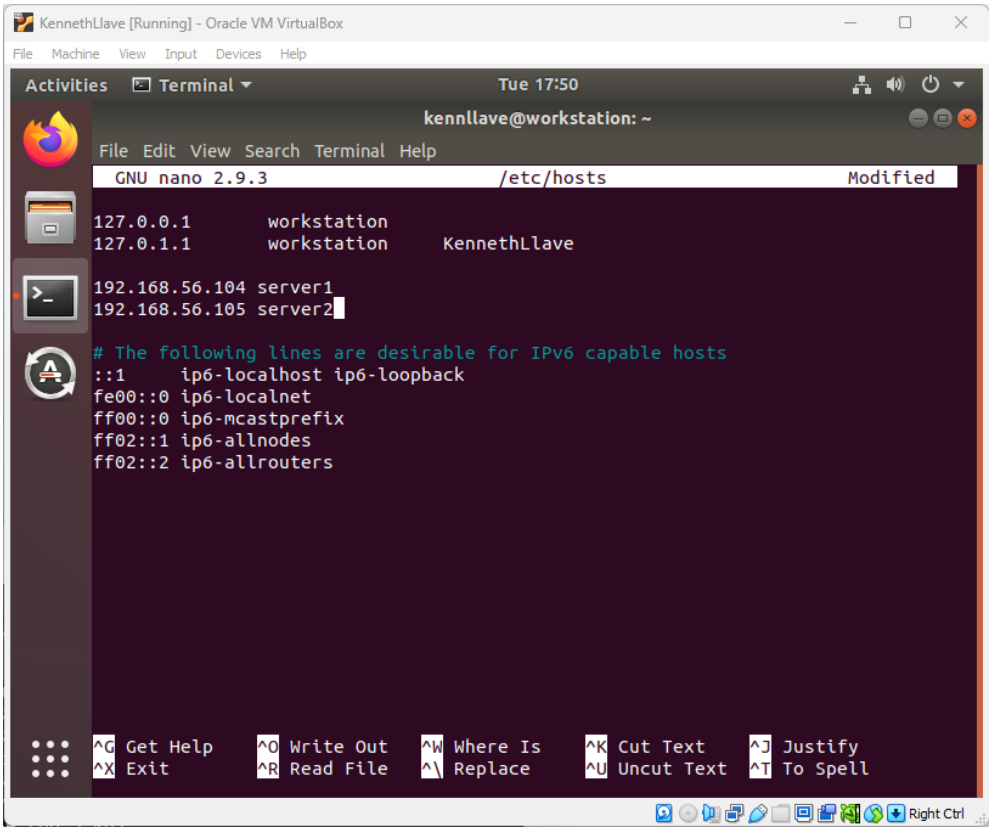




4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

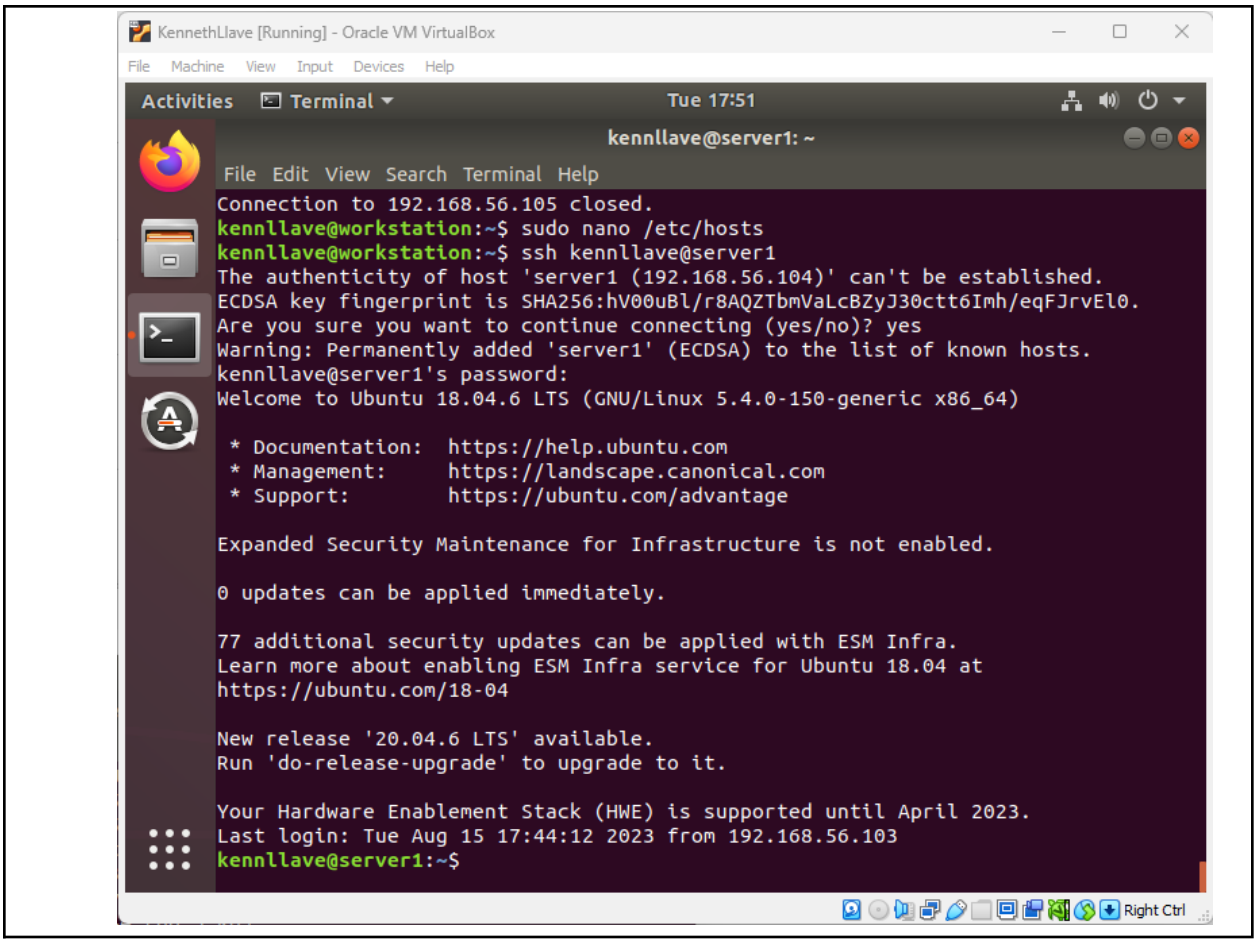
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

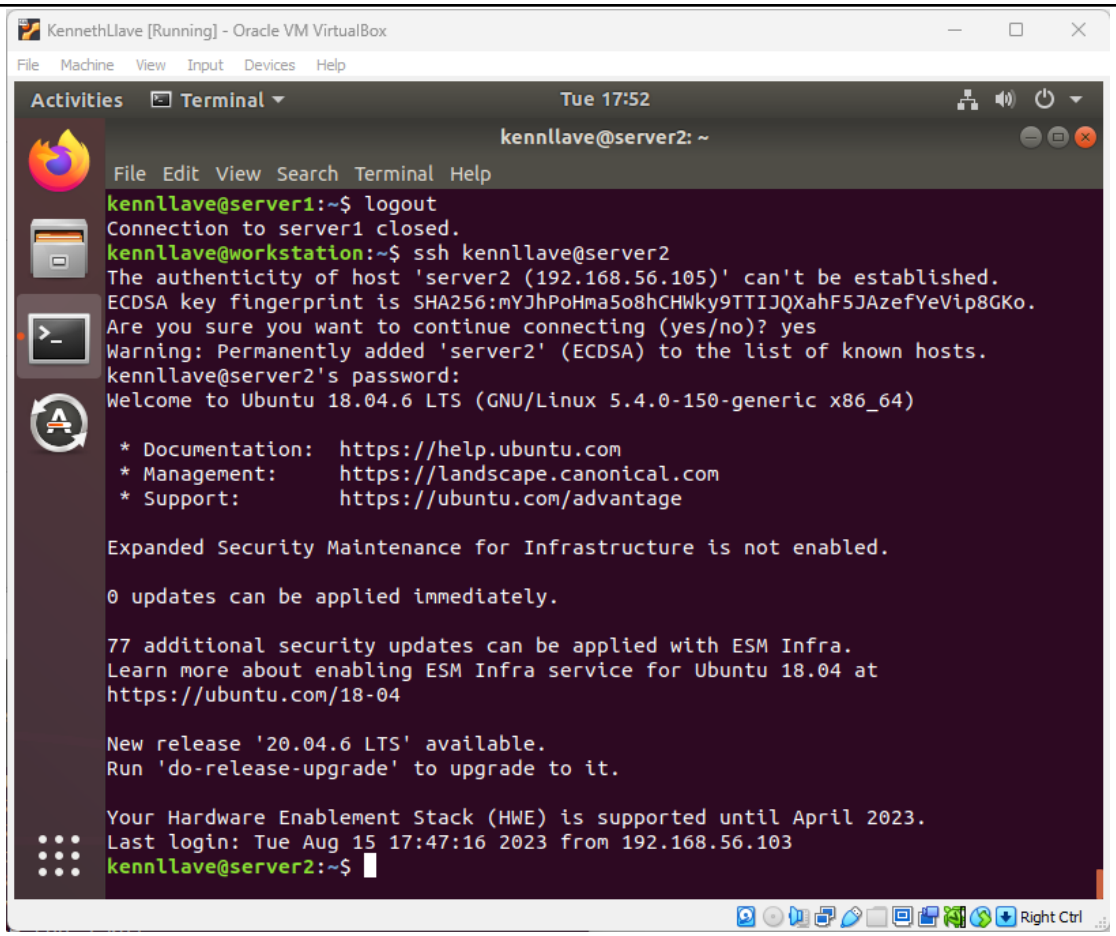
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)



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.


KennethLlave [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 17:51
kennllave@server1: ~
File Edit View Search Terminal Help
Connection to 192.168.56.105 closed.
kennllave@workstation:~$ sudo nano /etc/hosts
kennllave@workstation:~$ ssh kennllave@server1
The authenticity of host 'server1 (192.168.56.104)' can't be established.
ECDSA key fingerprint is SHA256:hV00uBl/r8AQZTbmVaLcBZyJ30ctt6Imh/eqFJrvEl0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1' (ECDSA) to the list of known hosts.
kennllave@server1's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
Expanded Security Maintenance for Infrastructure is not enabled.
0 updates can be applied immediately.
77 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04
New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Tue Aug 15 17:44:12 2023 from 192.168.56.103
kennllave@server1:~$
Right Ctrl



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   we were able to use the hostname instead of the IP address in SSH command because we edited the hosts file in /etc/hosts and we stated there what the IP address for servers 1 and 2 are. With this, our device can now correlate the IP addresses to their hostnames.
2. How secured is SSH?
   - SSH is secured because of its authentication process. Unlike in other authentication processes, SSH uses a generated **key**, with the key, you can enter without even knowing the password, but the key itself is encrypted and hard to crack, so it is still better than plain passwords and usernames.