Name: Pastrana, Mark Laurenz S.	Date Performed: Aug 22, 2023
Course/Section: CPE31S5	Date Submitted: Aug 23, 2023
Instructor: Engr. Richard Roman	Semester and SY: 1st, 2023-2024

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

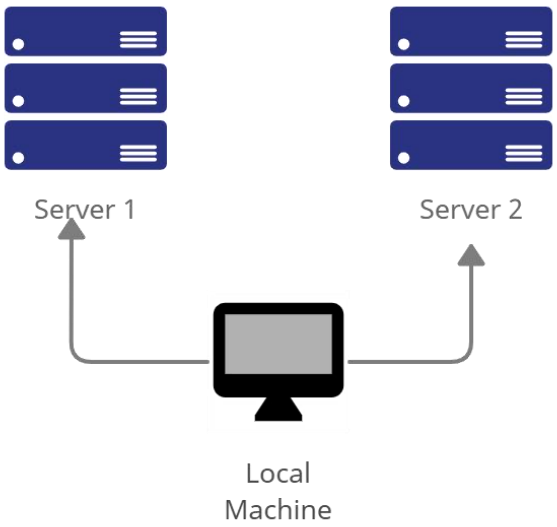
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

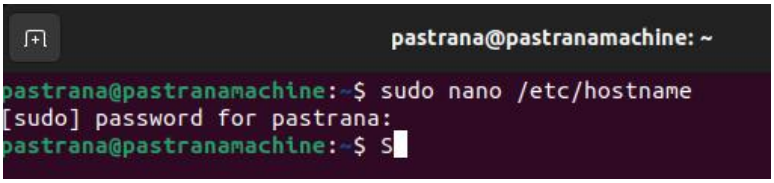
Network Topology:

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

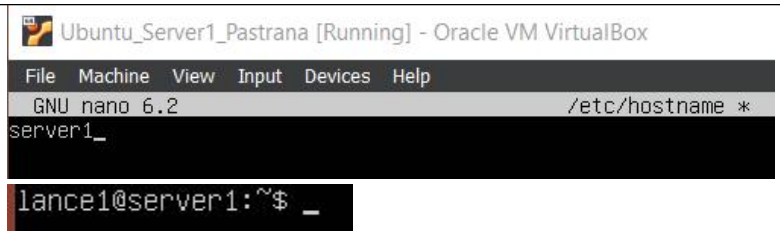


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

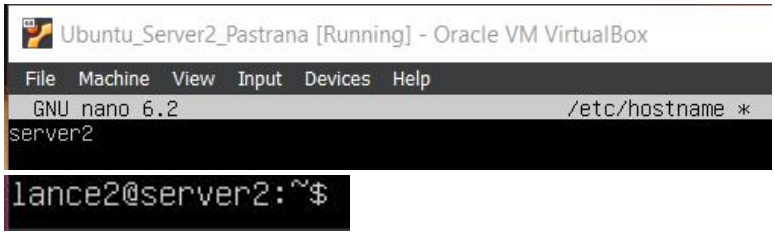
1. Change the hostname using the command *sudo nano /etc/hostname*



1.1 Use server1 for Server 1



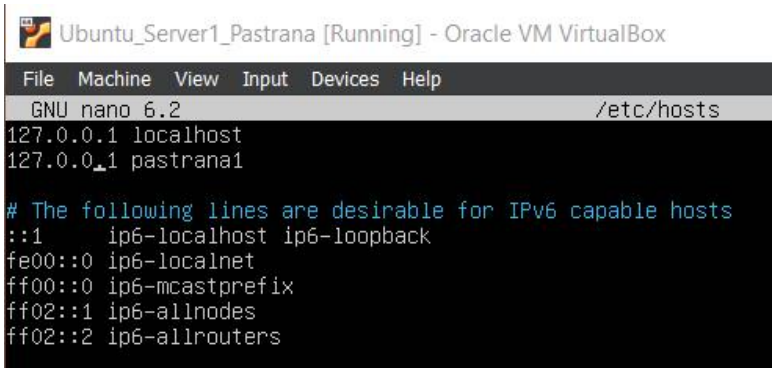
1.2 Use server2 for Server 2



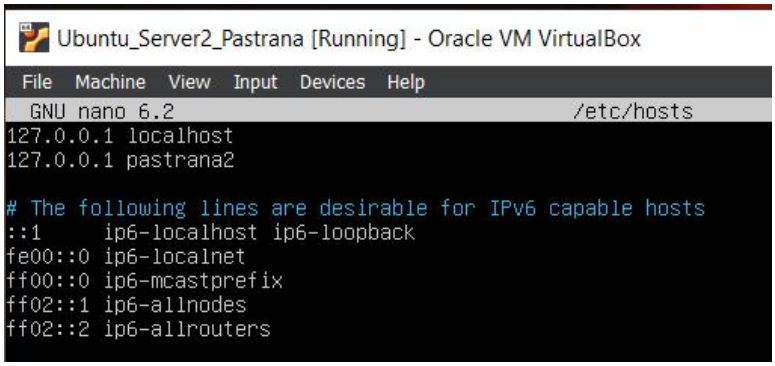
1.3 Use workstation for the Local Machine

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

```
pastrana@localmachine: ~
 GNU nano 6.2                        /etc/hosts
127.0.0.1       localhost
127.0.0.1       pastranamachine.myguest.virtualbox.org  pastranamachine

# 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
```

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.

server1 update

```
lance@pastranaserver1:~$ sudo apt update
[sudo] password for lance:
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/multiverse Sources [181 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [337 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe Sources [9,051 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/restricted Sources [32.4 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse Sources [12.2 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main Sources [301 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [467 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1,317 kB]
31% [7 Sources 4,771 kB/9,051 kB 53%] [12 Packages 760 kB/1,317 kB 58%]_
```

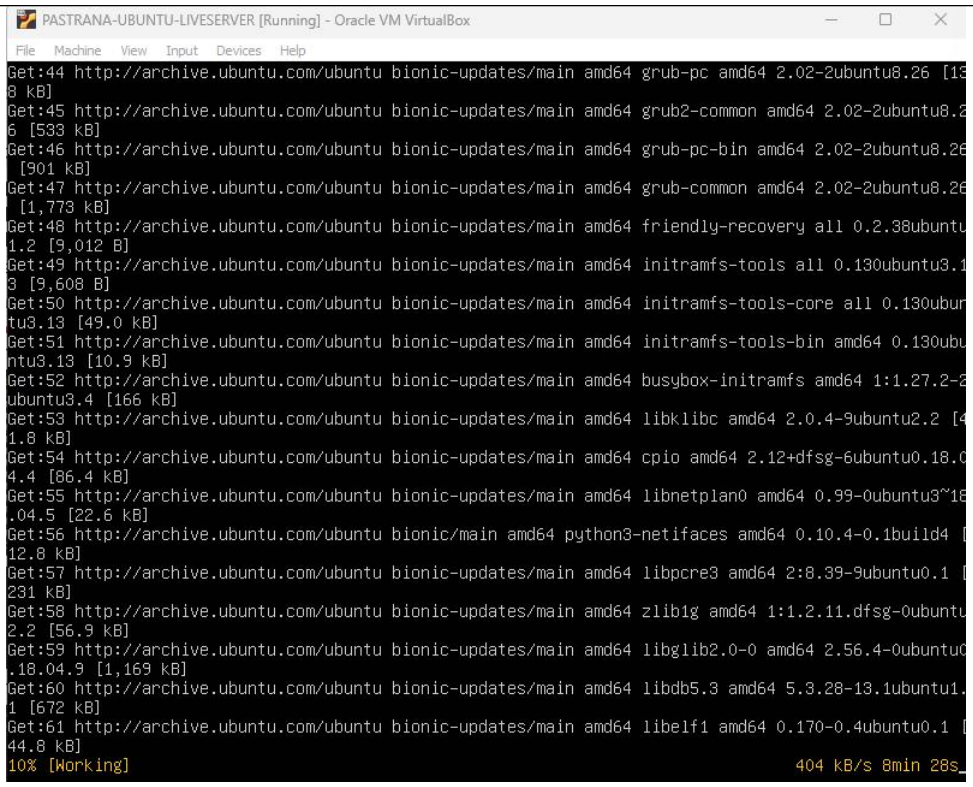
```
Fetched 34.6 MB in 31s (1,104 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
317 packages can be upgraded. Run 'apt list --upgradable' to see them.
lance@pastranaserver1:~$ _
```

server2 update

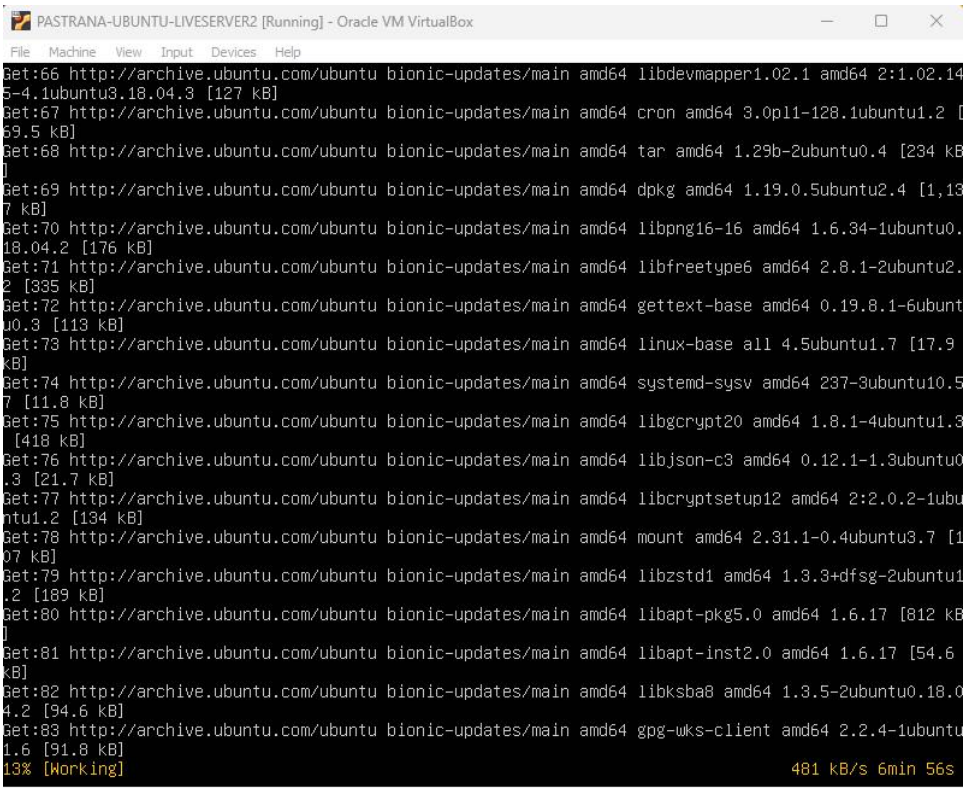
```
lance2@pastranaserver2:~$ sudo apt update
[sudo] password for lance2:
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security/main Sources [301 kB]
Hit:3 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/universe Sources [9,051 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/multiverse Sources [12.2 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [337 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted Sources [32.4 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [467 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1,317 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [182 kB]
32% [6 Sources 5,165 kB/9,051 kB 57%] [12 Translation-en 14.4 kB/182 kB 8%]          1,119 kB/s 23s
```

```
Fetched 34.4 MB in 32s (1,077 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
317 packages can be upgraded. Run 'apt list --upgradable' to see them.
lance2@pastranaserver2:~$
```

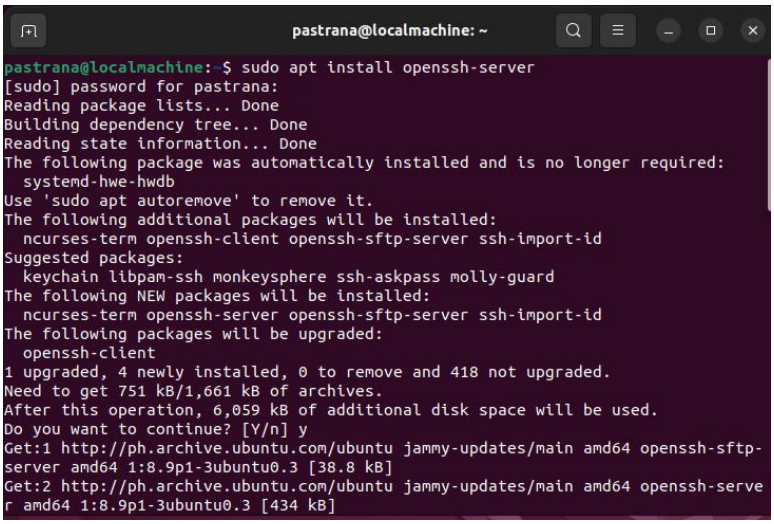
server1 upgrade



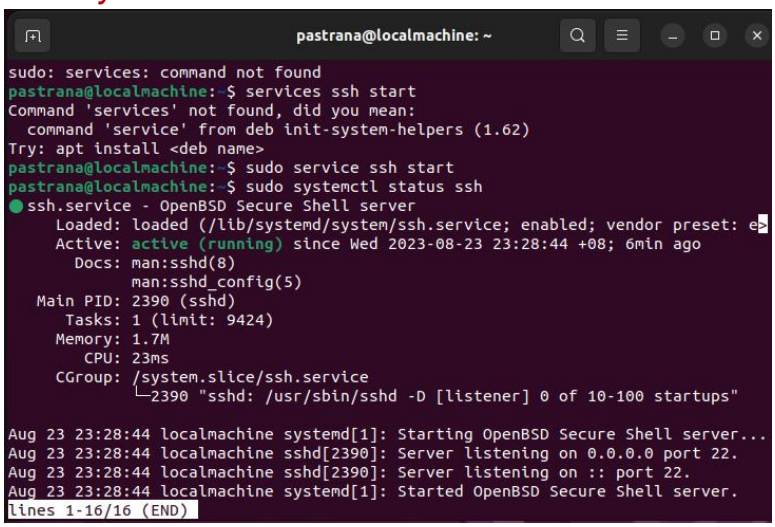
server2 upgrade

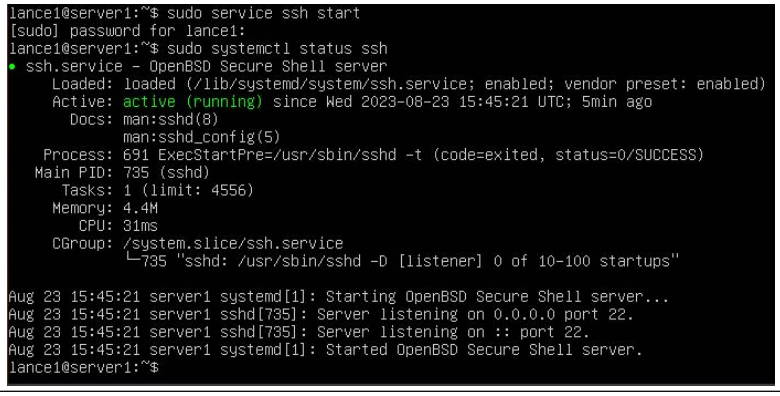


2. Install the SSH server using the command *sudo apt install openssh-server*.



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*





```
lance2@server2:~$ sudo service ssh start
[sudo] password for lance2:
lance2@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-08-23 15:46:17 UTC; 1min 35s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 688 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 727 (sshd)
      Tasks: 1 (limit: 5154)
     Memory: 4.4M
        CPU: 25ms
     CGroup: /system.slice/ssh.service
             └─727 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 23 15:46:17 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 23 15:46:17 server2 sshd[727]: Server listening on 0.0.0.0 port 22.
Aug 23 15:46:17 server2 sshd[727]: Server listening on :: port 22.
Aug 23 15:46:17 server2 systemd[1]: Started OpenBSD Secure Shell server.
lance2@server2:~$
```

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*

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

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

pastrana@localmachine:~$
```

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 **Local Machine** IP address: 192.168.56.**103**
   1.2 **Server 1** IP address: 192.168.56.**101**
   1.3 **Server 2** IP address: 192.168.56.**102**
2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ☒ Successful ☐ Not Successful

```
pastrana@localmachine:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.653 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.542 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.569 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.457 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.316 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.342 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.391 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.421 ms
64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.338 ms
64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.390 ms
64 bytes from 192.168.56.101: icmp_seq=11 ttl=64 time=0.416 ms
```

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

```
pastrana@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=0.576 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.436 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.559 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.540 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.488 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.538 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.499 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.369 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.356 ms
```

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

```
lance1@server1:~$ 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=0.749 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.498 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.513 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.435 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.463 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.375 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.457 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*

1.2 Enter the password for server 1 when prompted

```
pastrana@localmachine:~$ ssh lance1@192.168.56.101
lance1@192.168.56.101's password:
Welcome to Ubuntu 22.04.1 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 04:40:42 PM UTC 2023

  System load:  0.0               Processes:               131
  Usage of /:   44.4% of 11.21GB  Users logged in:         1
  Memory usage: 6%                IPv4 address for enp0s3: 192.168.56.101
  Swap usage:   0%
```

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

```
Last login: Wed Aug 23 16:29:00 2023
lance1@server1:~$
```

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

```
Last login: Wed Aug 23 16:29:00 2023
lance1@server1:~$
logout
Connection to 192.168.56.101 closed.
pastrana@localmachine:~$
```

3. Do the same for Server 2.

```
pastrana@localmachine:~$ ssh lance2@192.168.56.102
lance2@192.168.56.102's password:
Welcome to Ubuntu 22.04.1 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 04:43:04 PM UTC 2023

  System load:  0.0029296875      Processes:              132
  Usage of /:   25.9% of 11.21GB  Users logged in:        1
  Memory usage: 5%                IPv4 address for enp0s3: 192.168.56.102
```

```
Last login: Wed Aug 23 16:29:06 2023
lance2@server2:~$
```

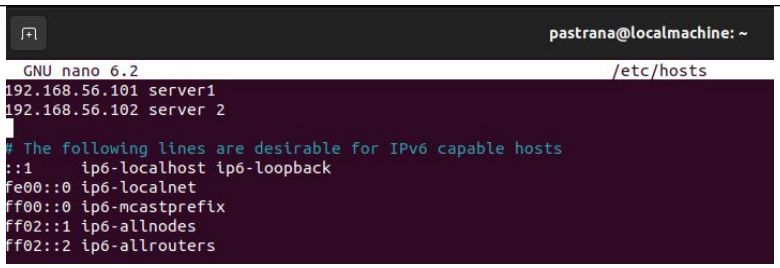
```
Last login: Wed Aug 23 16:29:06 2023
lance2@server2:~$
logout
Connection to 192.168.56.102 closed.
pastrana@localmachine:~$
```

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

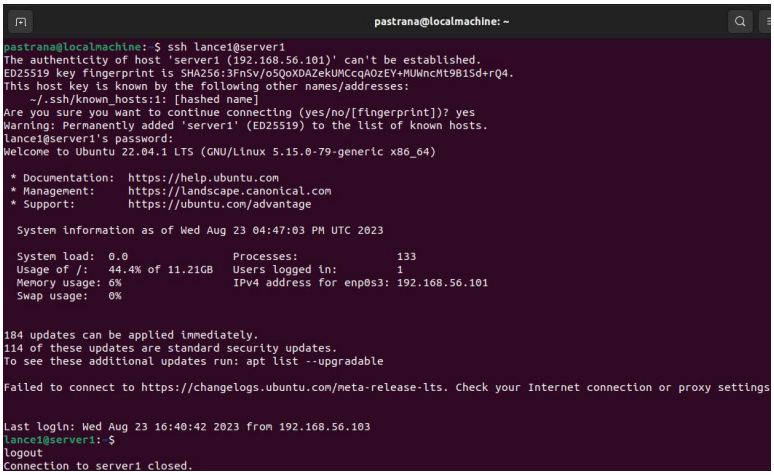
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

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

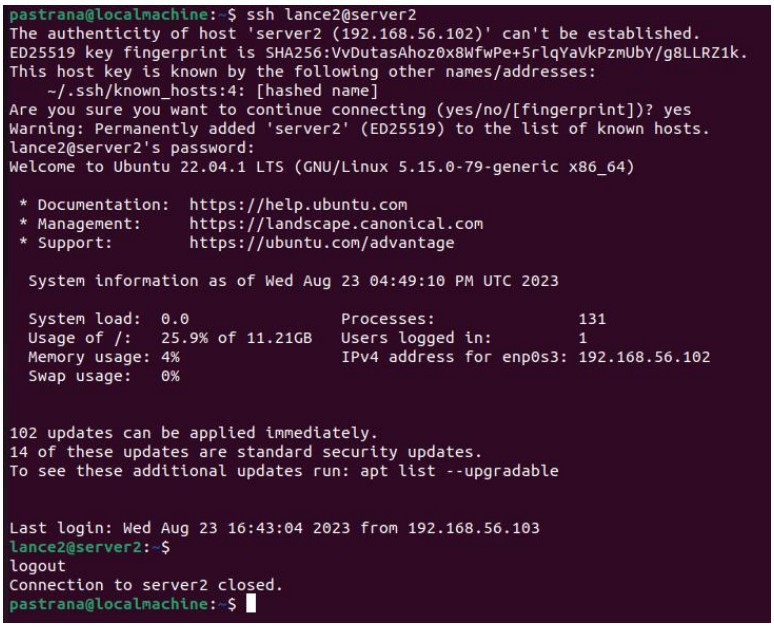
4.3 Save the file and exit.



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?
   By modifying the local machine's host file via the command 'sudo nano /etc/hosts' and inputting the IP addresses of the respective servers alongside their hostnames, we achieved to utilize the server's hostname instead of the IP address in SSH commands. This adjustment allowed seamless translation between hostnames and IP addresses, giving us accessibility and streamlining SSH connections.

2. How secure is SSH?

   Due to its encryption, authentication techniques such as public keys, and strong key exchange protocols, SSH is known as very secure for remote access and data transfer. Its security, however, is dependent on correct configuration, strong password restrictions, and up-to-date software. SSH, when used properly, provides a reliable and secure method of connecting to remote systems.