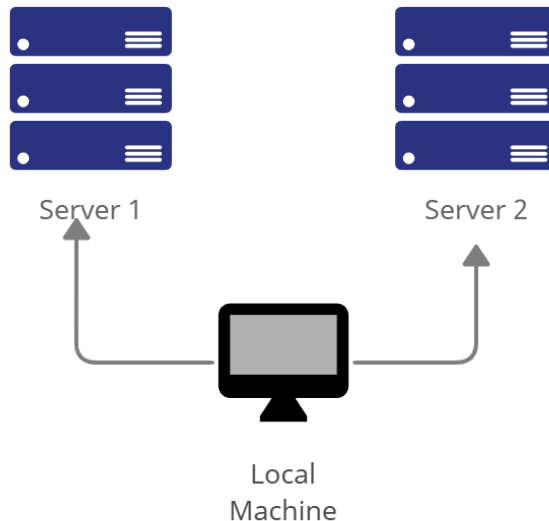
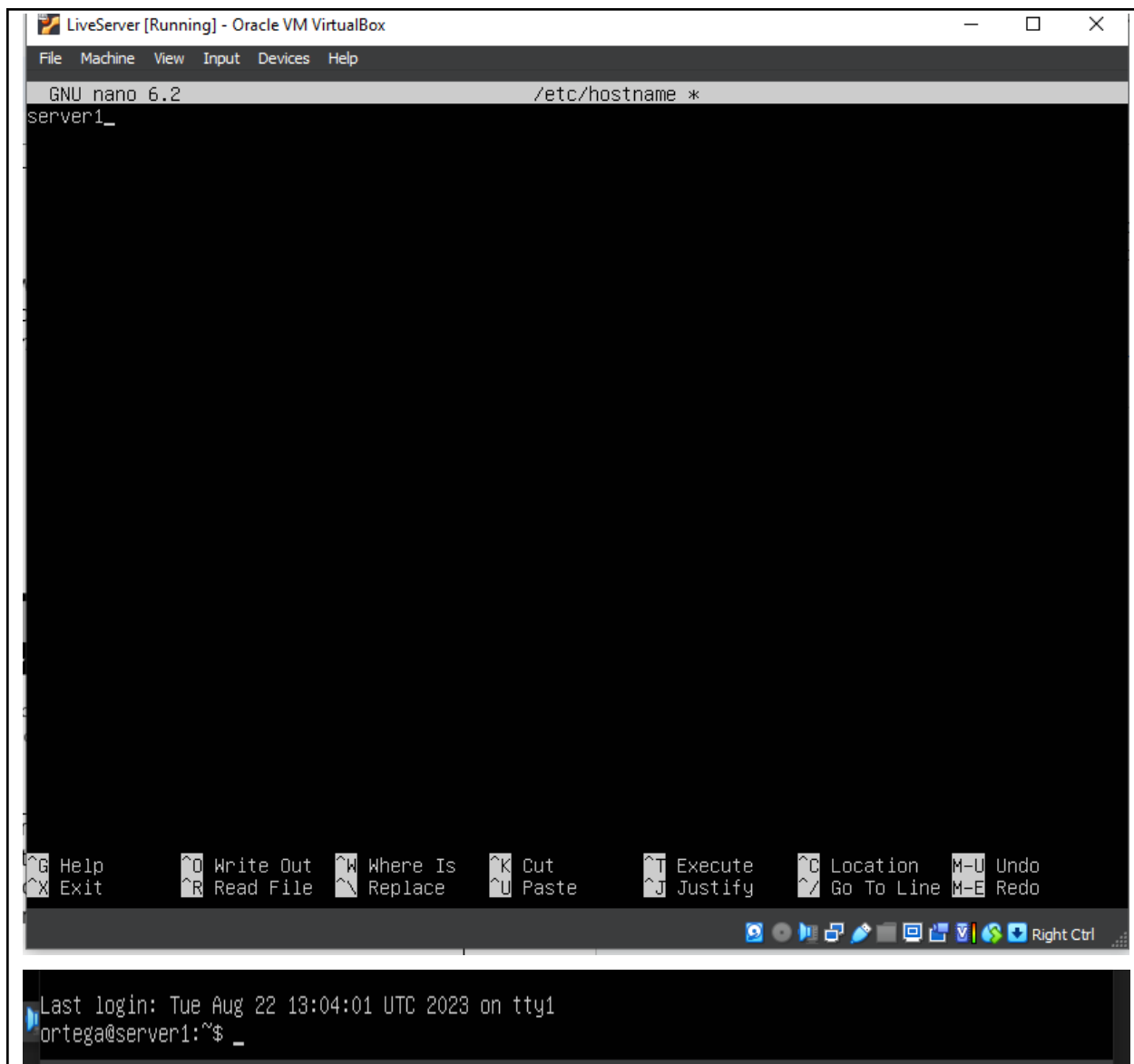
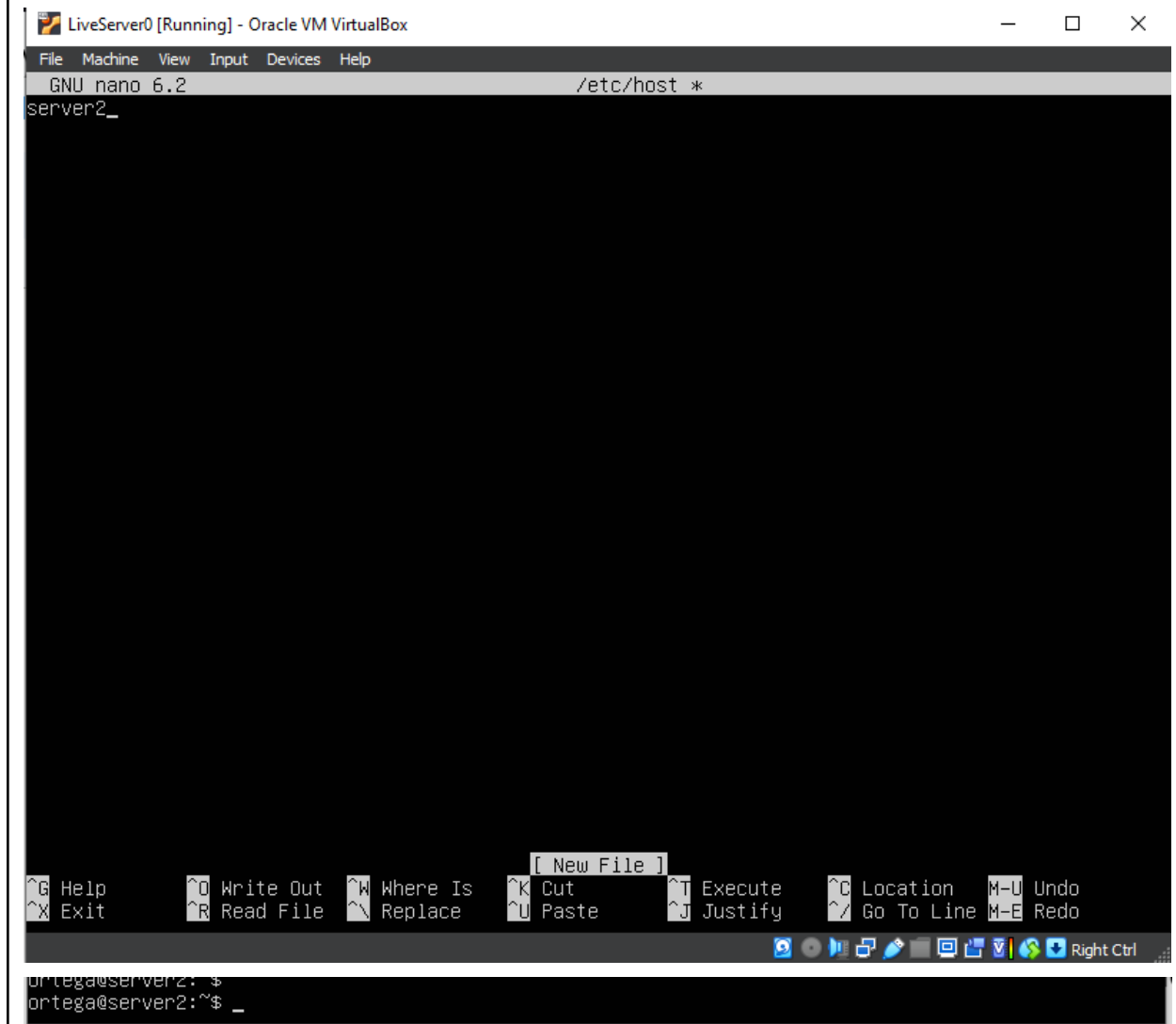


Name: Victor B. Ortega	Date Performed: 08/22/23
Course/Section: CPE232	Date Submitted: 08/23/23
Instructor: Engr Roman Richard	Semester and SY: 2023-2024
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 1.2. 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, <i>provide screenshots for each task</i> . (Note: <i>it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine</i>).	
 <pre> graph TD LocalMachine[Local Machine] --> Server1[Server 1] LocalMachine --> Server2[Server 2] </pre> <p>The diagram illustrates a network topology. At the bottom center is a computer icon labeled "Local Machine". Two lines extend upwards from the "Local Machine" to two server stacks. The left server stack is labeled "Server 1" and the right server stack is labeled "Server 2". Each server stack consists of three blue rectangular blocks, representing server racks.</p>	
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 <i>sudo nano /etc/hostname</i> 1.1 Use server1 for Server 1	



1.2 Use server2 for Server 2

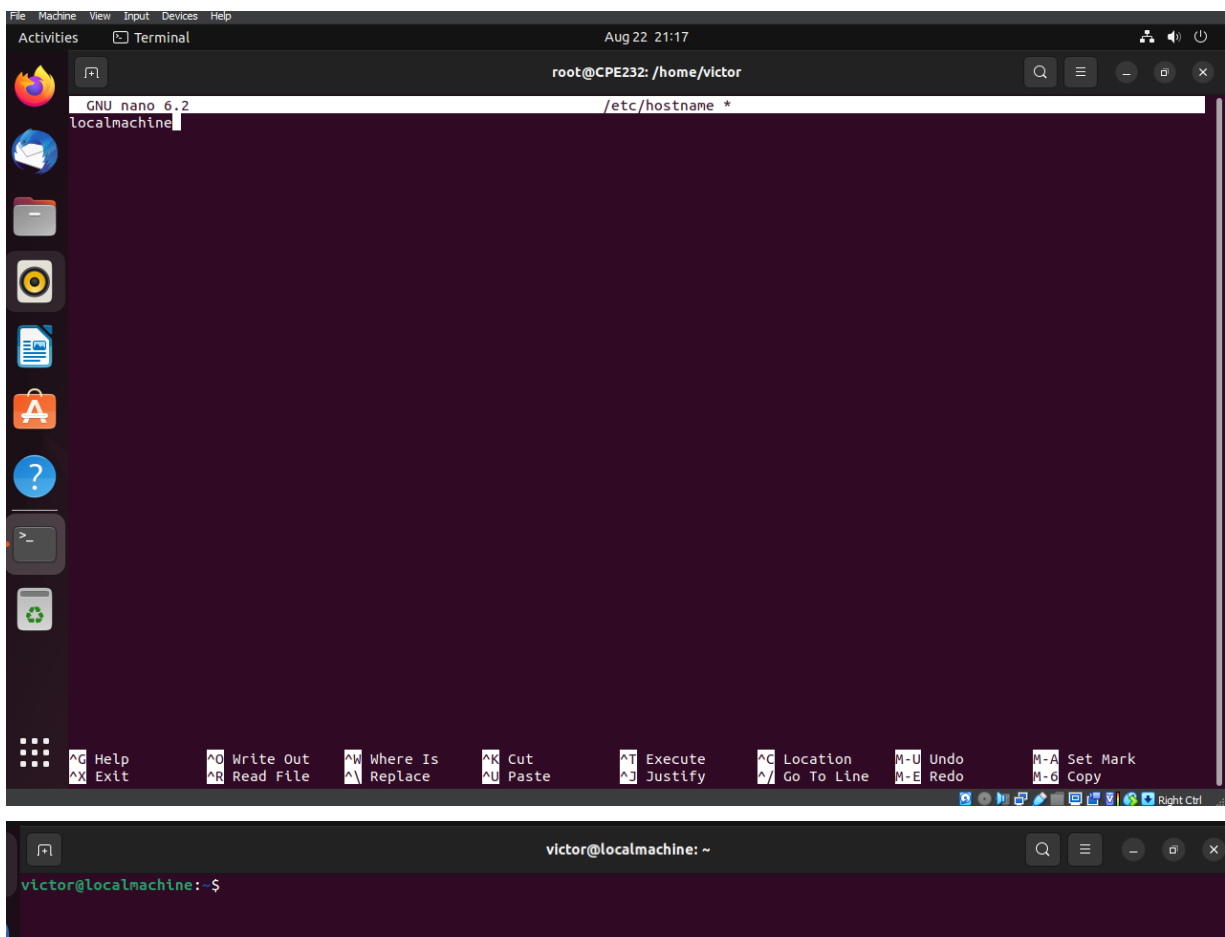


```
LiveServer0 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2 /etc/host *
server2_

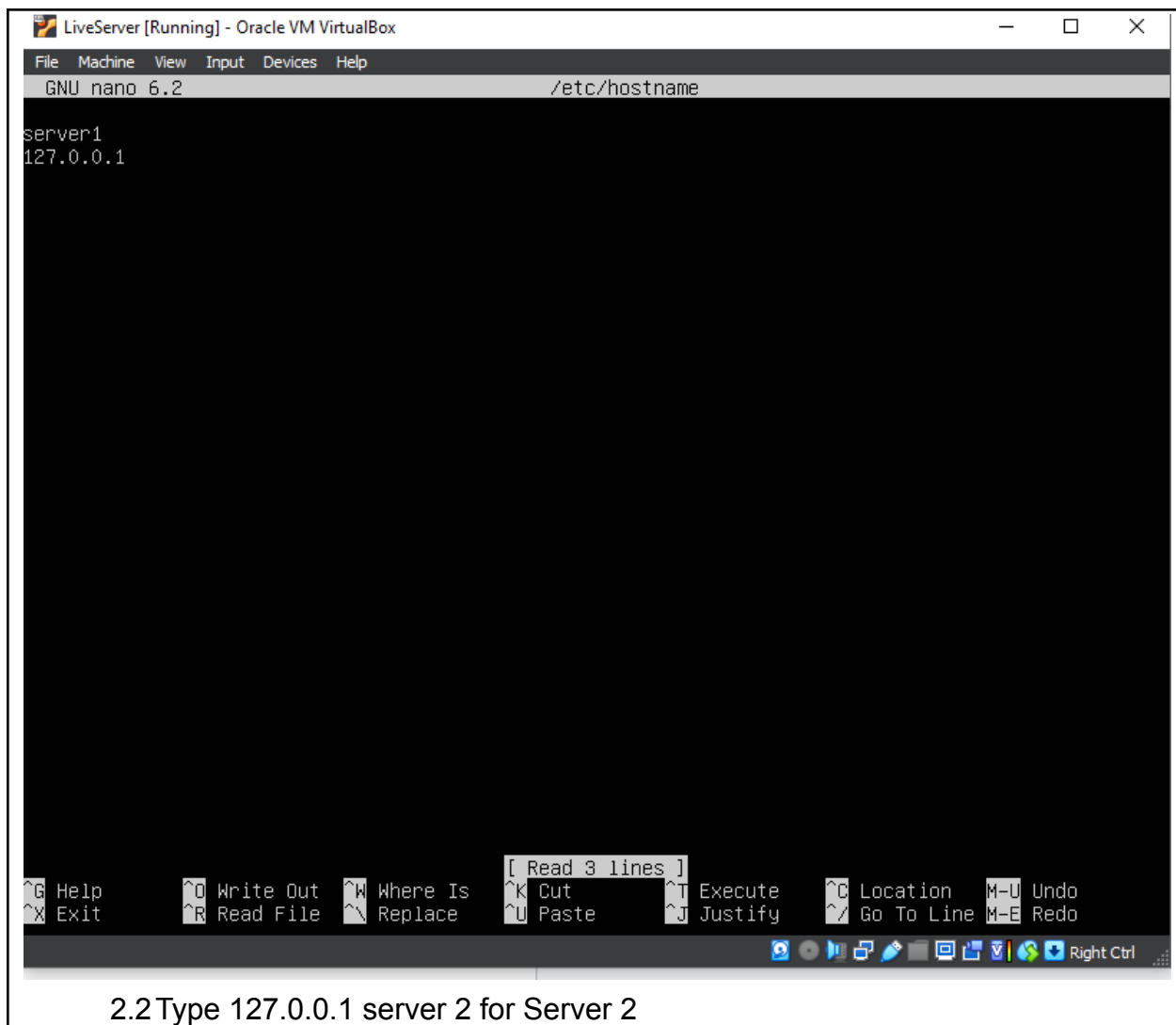
[ New File ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo

ortega@server2:~$
ortega@server2:~$ _
```

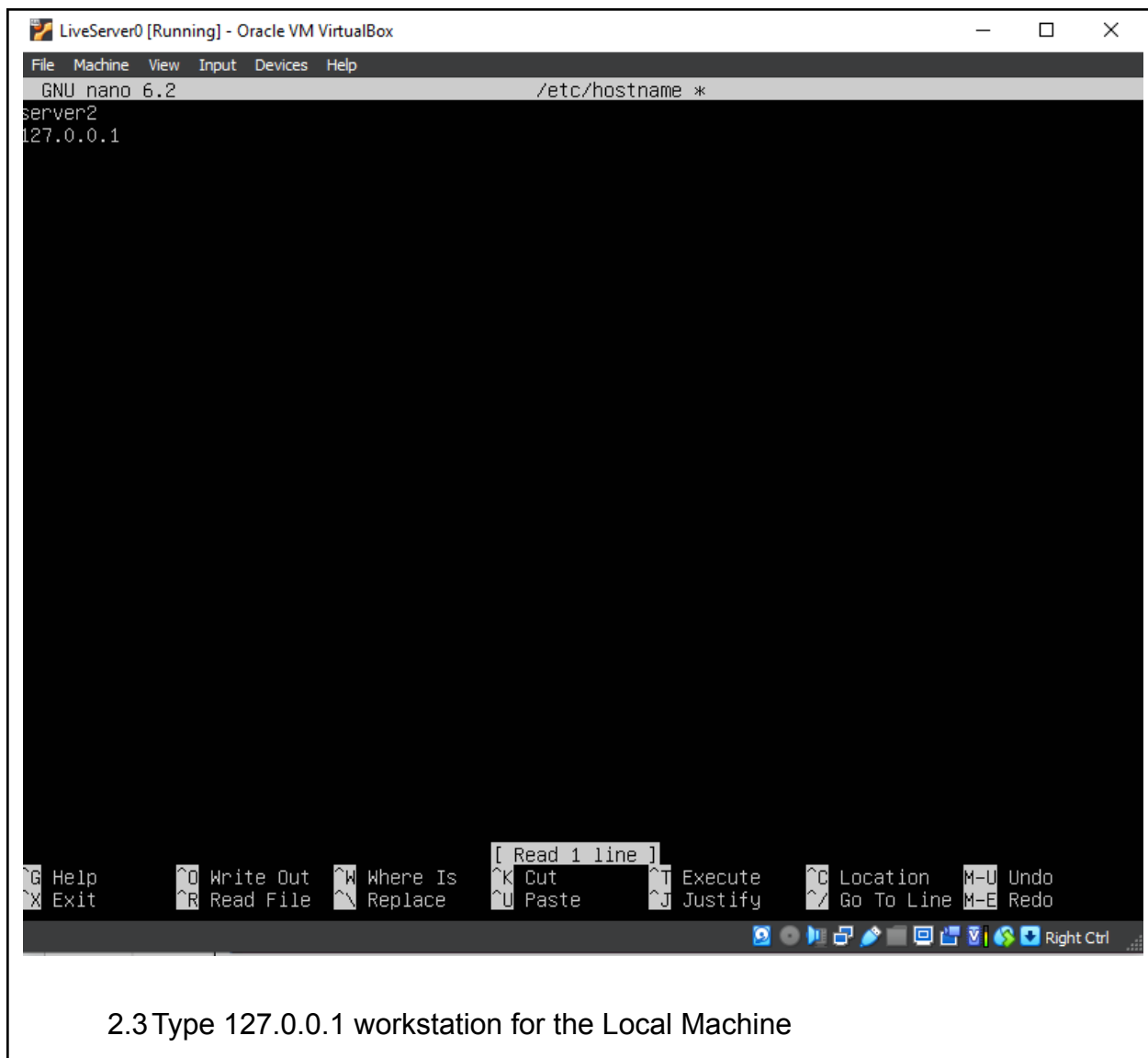
1.3 Use workstation for the Local Machine

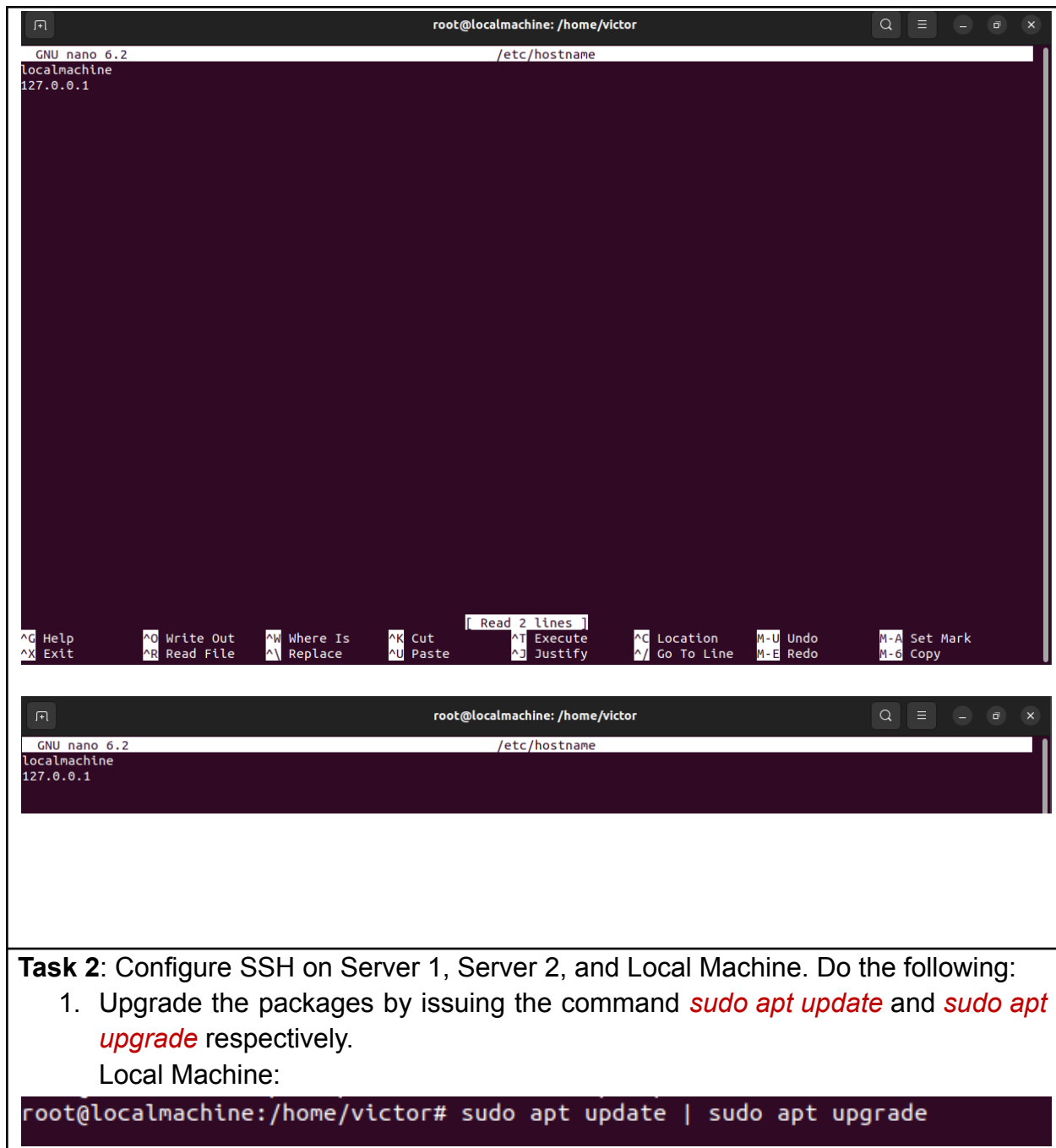


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
 - 2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2





```
root@localmachine: /home/victor
GNU nano 6.2 /etc/hostname
localmachine
127.0.0.1

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/_ Go To Line  M-E Redo      M-G Copy

[ Read 2 lines ]

root@localmachine: /home/victor
GNU nano 6.2 /etc/hostname
localmachine
127.0.0.1

root@localmachine: /home/victor# sudo apt update | sudo apt upgrade
```

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.

Local Machine:

```

Preparing to unpack .../20-libldap-common_2.5.16+dfsg-0ubuntu0.22.04.1_all.deb ...
Unpacking libldap-common (2.5.16+dfsg-0ubuntu0.22.04.1) over (2.5.15+dfsg-0ubuntu0.22.04.1) ...
Preparing to unpack .../21-intel-microcode_3.20230808.0ubuntu0.22.04.1_amd64.deb ...
Unpacking intel-microcode (3.20230808.0ubuntu0.22.04.1) over (3.20230214.0ubuntu0.22.04.1) ...
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.4) ...
Setting up apt-utils (2.4.10) ...
Setting up intel-microcode (3.20230808.0ubuntu0.22.04.1) ...
update-initramfs: deferring update (trigger activated)
intel-microcode: microcode will be updated at next boot
Setting up libjavascriptcoregtk-4.0-18:amd64 (2.40.5-0ubuntu0.22.04.1) ...
Setting up gir1.2-javascriptcoregtk-4.0:amd64 (2.40.5-0ubuntu0.22.04.1) ...
Setting up libldap-common (2.5.16+dfsg-0ubuntu0.22.04.1) ...
Setting up libwbclient0:amd64 (2:4.15.13+dfsg-0ubuntu1.3) ...
Setting up libldap-2.5-0:amd64 (2.5.16+dfsg-0ubuntu0.22.04.1) ...
Setting up xxd (2:8.2.3995-1ubuntu2.11) ...
Setting up vim-common (2:8.2.3995-1ubuntu2.11) ...
Setting up libwebkit2gtk-4.0-37:amd64 (2.40.5-0ubuntu0.22.04.1) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu0.5) ...
Setting up initramfs-tools-bin (0.140ubuntu13.4) ...
Setting up gir1.2-webkit2-4.0:amd64 (2.40.5-0ubuntu0.22.04.1) ...
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.4) ...
Setting up samba-ls:amd64 (2:4.15.13+dfsg-0ubuntu1.3) ...
Setting up vim-tiny (2:8.2.3995-1ubuntu2.11) ...
Setting up ghostscript (9.55.0~dfsg1-0ubuntu5.4) ...
Setting up libsmbclient:amd64 (2:4.15.13+dfsg-0ubuntu1.3) ...
Setting up initramfs-tools-core (0.140ubuntu13.4) ...
Setting up initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: deferring update (trigger activated)
Setting up ghostscript-x (9.55.0~dfsg1-0ubuntu5.4) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.2.0-26-generic
root@localmachine:/home/victor#

```

Server 1:

```

root@server1:/home/ortega# sudo apt update | sudo apt upgrade -y

```

```

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server1:/home/ortega#

```

Server2:

```

[sudo] password for ortega:
root@server2:/home/ortega# sudo apt update | sudo apt upgrade -y

```

```

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done

```


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

Local Machine:

```
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:tvDEBWZi/vyHImon9ClVVvpdaLYn0mFxz9cAR65mJQc root@localmachine (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:ypH3gbqKSI0FuvooHj3j3HShesxy0e6bkgeH8yB3IUo root@localmachine (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:2llwmpj6kFvLYl8ilNxRns2r/aQqIFxdHOPAAuqTQ+w root@localmachine (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
```

Server 1:

```
root@server1:/home/ortega# sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server1:/home/ortega#
```

Server 2:

```
root@server2:/home/ortega# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server2:/home/ortega#
```

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

3.1 *sudo service ssh start*

Local Machine:

```
root@localmachine:/home/victor# sudo service ssh start
root@localmachine:/home/victor#
```

Server 1:

```
root@server1:/home/ortega# sudo service ssh start
root@server1:/home/ortega#
```

Server 2:

```
root@server2:/home/ortega# sudo service ssh start
root@server2:/home/ortega#
```

3.2 *sudo systemctl status ssh*

Local Machine:

```
root@localmachine:/home/victor# 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 Tue 2023-08-22 21:48:27 +08; 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 11477 (sshd)
    Tasks: 1 (limit: 5655)
   Memory: 1.7M
      CPU: 18ms
   CGroup: /system.slice/ssh.service
           └─11477 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 21:48:26 localmachine systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 21:48:27 localmachine sshd[11477]: Server listening on 0.0.0.0 port 22.
Aug 22 21:48:27 localmachine sshd[11477]: Server listening on :: port 22.
Aug 22 21:48:27 localmachine systemd[1]: Started OpenBSD Secure Shell server.
```

Server 1:

```
root@server1:/home/ortega# sudo service ssh start
root@server1:/home/ortega# 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 Tue 2023-08-22 13:42:27 UTC; 14min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 672 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 711 (sshd)
    Tasks: 1 (limit: 4908)
   Memory: 4.4M
      CPU: 22ms
   CGroup: /system.slice/ssh.service
           └─711 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 13:42:26 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 13:42:27 server1 sshd[711]: Server listening on 0.0.0.0 port 22.
Aug 22 13:42:27 server1 sshd[711]: Server listening on :: port 22.
Aug 22 13:42:27 server1 systemd[1]: Started OpenBSD Secure Shell server.
root@server1:/home/ortega#
```

Server 2:

```
root@server2:/home/ortega# sudo service ssh start
root@server2:/home/ortega# 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 Tue 2023-08-22 13:44:45 UTC; 12min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 680 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 723 (sshd)
      Tasks: 1 (limit: 4557)
     Memory: 4.4M
        CPU: 21ms
    CGroup: /system.slice/ssh.service
            └─723 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 13:44:44 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 13:44:45 server2 sshd[723]: Server listening on 0.0.0.0 port 22.
Aug 22 13:44:45 server2 sshd[723]: Server listening on :: port 22.
Aug 22 13:44:45 server2 systemd[1]: Started OpenBSD Secure Shell server.
root@server2:/home/ortega# _
```

4. Configure the firewall to all port 22 by issuing the following commands:

4.1 *sudo ufw allow ssh*

Local Machine:

```
root@localmachine:/home/victor# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@localmachine:/home/victor#
```

Server 1:

```
root@server1:/home/ortega# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@server1:/home/ortega#
```

Server 2:

```
root@server2:/home/ortega# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@server2:/home/ortega# _
```

4.2 *sudo ufw enable*

Local Machine:

```
root@localmachine:/home/victor# sudo ufw enable
Firewall is active and enabled on system startup
root@localmachine:/home/victor#
```

Server 1:

```
root@server1:/home/ortega# sudo ufw enable
Firewall is active and enabled on system startup
root@server1:/home/ortega#
```

Server 2:

```
rules updated (v6)
root@server2:/home/ortega# sudo ufw enable
Firewall is active and enabled on system startup
root@server2:/home/ortega#
```

4.3 *sudo ufw status*

Local Machine:

```
victor@workstation:~$ sudo su
root@workstation:/home/victor# sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
root@workstation:/home/victor#
```

Server 1:

```
root@server1:/home/ortega# sudo ufw status
Status: active

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

Server 2:

```
root@server2:/home/ortega# sudo ufw status
Status: active

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

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: 192.168.56.101

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
```

1.2 Server 2 IP address: 192.168.56.103

```
enp0s3: 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::a00:27ff:fe25:5c90 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:25:5c:90 txqueuelen 1000 (Ethernet)
```

1.3 Server 3 IP address: 192.168.56.102

```
root@server2:/home/ortega# ifconfig
enp0s3: 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::a00:27ff:fefa:cce1 prefixlen 64 scopeid 0x20<link>
```

2. Make sure that they can ping each other.

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

```
victor@workstation:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.438 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.269 ms
```

```
^Z
[1]+  Stopped                  ping 192.168.56.103
victor@workstation:~$
```

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

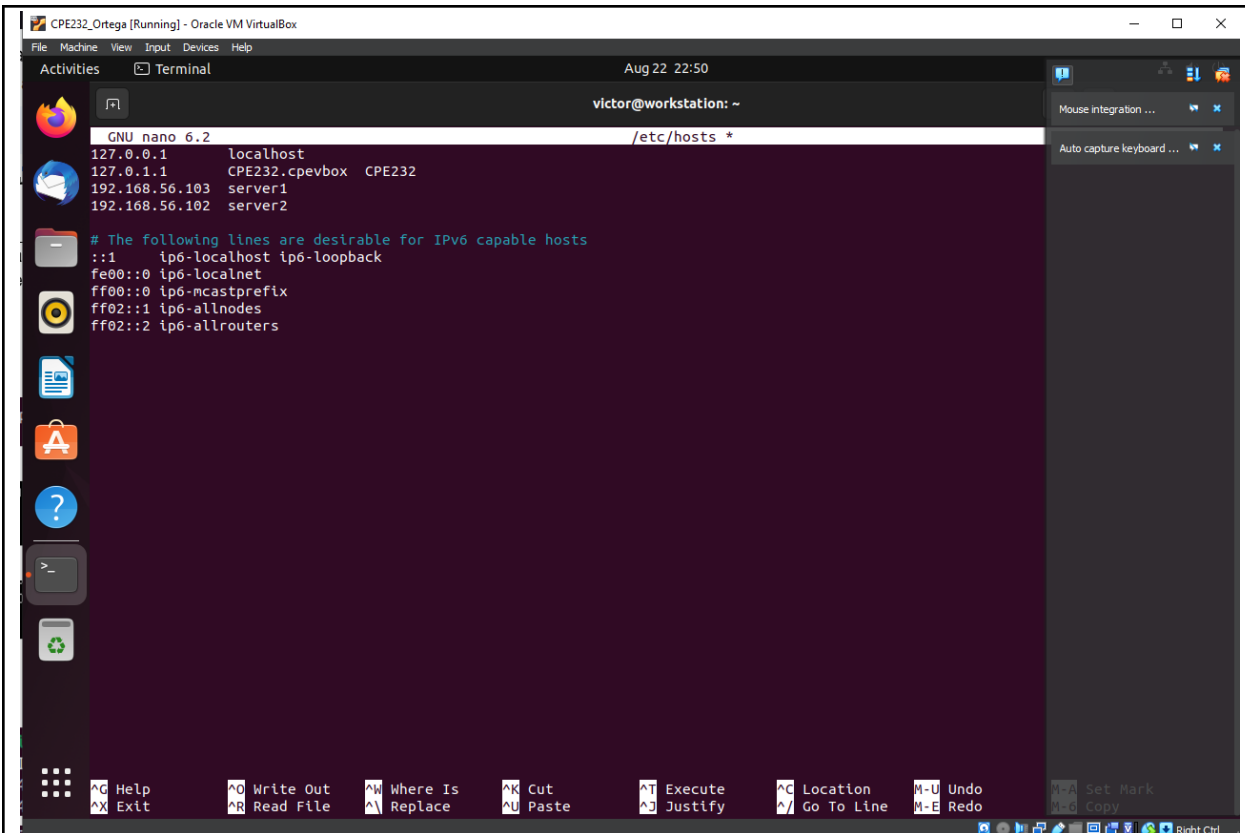
```
[1]+  Stopped                  ping 192.168.56.103
victor@workstation:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.374 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.265 ms
^Z
[2]+  Stopped                  ping 192.168.56.102
victor@workstation:~$
```

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

```
root@server1:/home/ortega# 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=0.417 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.273 ms
^Z
[1]+  Stopped                  ping 192.168.56.102
root@server1:/home/ortega# _
```

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 jvtaylor@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, `jvtaylor@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)

A screenshot of an Oracle VM VirtualBox window titled 'CPE232_Ortega [Running] - Oracle VM VirtualBox'. The window shows a terminal window with the prompt 'victor@workstation: ~'. The terminal is running the 'nano' text editor, editing the file '/etc/hosts'. The content of the file is as follows:

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 CPE232.cpevbox CPE232
192.168.56.103 server1
192.168.56.102 server2

# 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
```

The terminal window has a dark background with light-colored text. The nano editor's status bar at the bottom shows various keyboard shortcuts like 'Help', 'Exit', 'Write Out', 'Read File', etc. The VirtualBox window also shows a menu bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help'. On the right side of the terminal window, there are two small panels: 'Mouse integration ...' and 'Auto capture keyboard ...'. The bottom of the VirtualBox window shows a taskbar with various icons and a system tray on the right with a 'Right Ctrl' button.

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 jvtaylor@server1`. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

Server 1:

```
victor@workstation:~$ ssh ortega@server1
ortega@server1's password:
Welcome to Ubuntu 22.04.3 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 Tue Aug 22 02:53:10 PM UTC 2023

System load:  0.0390625          Processes:            117
Usage of /:   44.6% of 11.21GB   Users logged in:     1
Memory usage: 5%                IPv4 address for enp0s3: 192.168.56.103
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Aug 22 14:34:03 2023 from 192.168.56.101
ortega@server1:~$
```


Server 2:

```
victor@workstation:~$ ssh ortega@server2
The authenticity of host 'server2 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:pXLJdDGMb1xCY35MP5UJXvN9svNfR/oCm2IvRNQ6AS4.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2' (ED25519) to the list of known hosts.
ortega@server2's password:
Welcome to Ubuntu 22.04.3 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 Tue Aug 22 02:54:31 PM UTC 2023

System load:  0.0               Processes:           116
Usage of /:   44.7% of 11.21GB   Users logged in:    1
Memory usage: 6%               IPv4 address for enp0s3: 192.168.56.102
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Aug 22 14:45:45 2023 from 192.168.56.101
ortega@server2:~$
```

(1-3)

Local Machine to Server 1:

Verifying:

```
Welcome to Ubuntu 22.04.3 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 Tue Aug 22 02:34:03 PM UTC 2023

System load:  0.0               Processes:            117
Usage of /:   44.6% of 11.21GB  Users logged in:     1
Memory usage: 5%               IPv4 address for enp0s3: 192.168.56.103
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Aug 22 14:23:42 2023
ortega@server1:~$
```

Log out:

```
logout
Connection to 192.168.56.103 closed.
victor@workstation:~$
```

Local Machine to Server 2:

Verifying:

```
Welcome to Ubuntu 22.04.3 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 Tue Aug 22 02:45:44 PM UTC 2023

System load:  0.0               Processes:            116
Usage of /:   44.7% of 11.21GB   Users logged in:     1
Memory usage: 5%               IPv4 address for enp0s3: 192.168.56.102
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Aug 22 14:22:12 2023
```

Log out:

```
logout
Connection to 192.168.56.102 closed.
victor@workstation:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
In task 4, we included the server IP address of server 1 and server 2 in “/etc/host”.
2. How secured is SSH?

Due to its robust encryption, authentication procedures, and configuration choices, SSH in Ubuntu is typically safe. However, maintaining its security calls for good key management, frequent updates, and adherence to best practices. Lastly, SSH can communicate to each other within the network.

Conclusion:

In conclusion, employing virtual machines to configure networks gives a flexible and effective alternative for contemporary IT systems. Through the use of virtual machines, organizations and individuals may mimic a variety of network scenarios without the requirement for specialized real hardware. This solution maintains the advantages of isolation, security, and simple management while promoting effective resource usage, cost savings, and quick provisioning of network configurations. Virtual machine-based network setup accelerates procedures, improves experimentation, and eventually helps to the growth of network management methods in an increasingly dynamic digital environment, whether used for testing, development, training, or production reasons.

Honor Pledge

“ I affirm that I shall not give or receive any unauthorized help on this assignment and that all work is my own”

