Hands-on Activity 0 - Creating Virtual Machines	
Name: Torrecampo, Juan Piolo S.	**Date Performed:** August 16, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** August 18, 2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Sem, 2022 - 2023

1. Objectives:

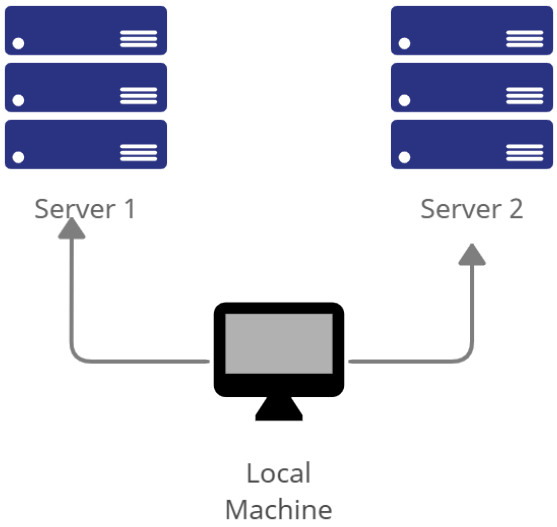
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. Discussion:

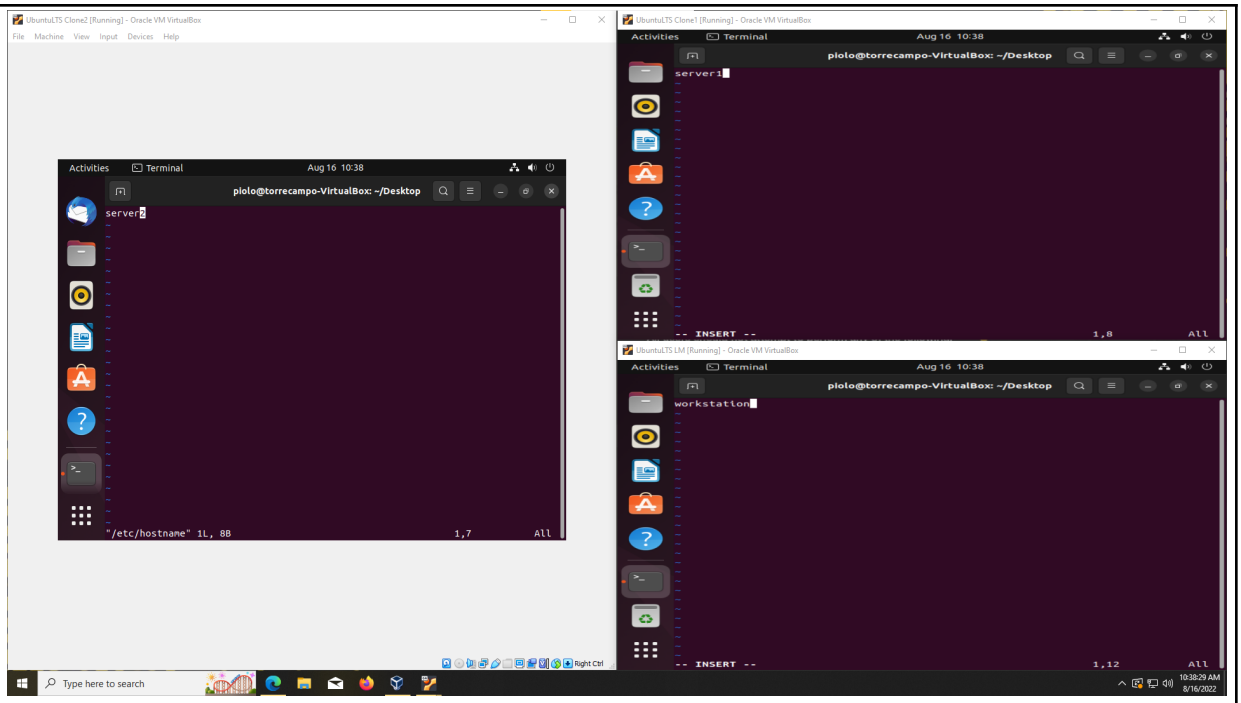
Network Topology:

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
   1.2 Use server1 for Server 1
   1.3 Use server2 for Server 2
   1.4 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

```
127.0.0.1       localhost
127.0.1.1       server1

# 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
```

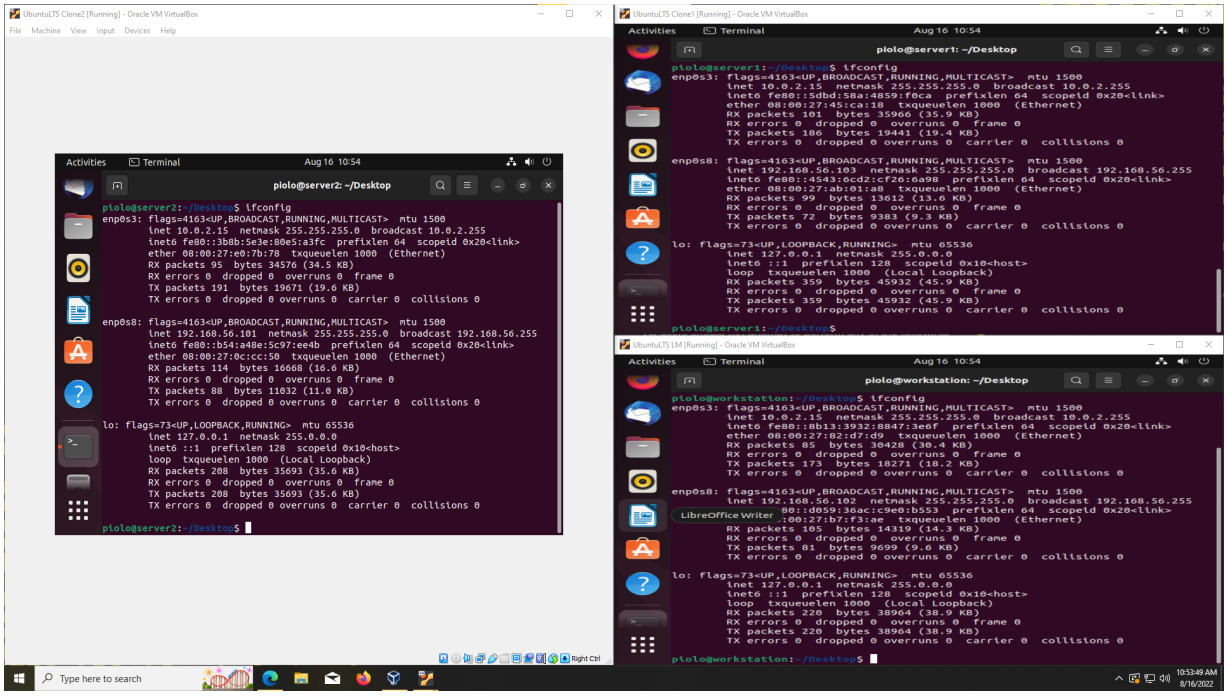
2.2 Type 127.0.0.1 server 2 for Server 2

```
127.0.0.1       localhost
127.0.1.1       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
```

2.3 Type 127.0.0.1 workstation for the Local Machine

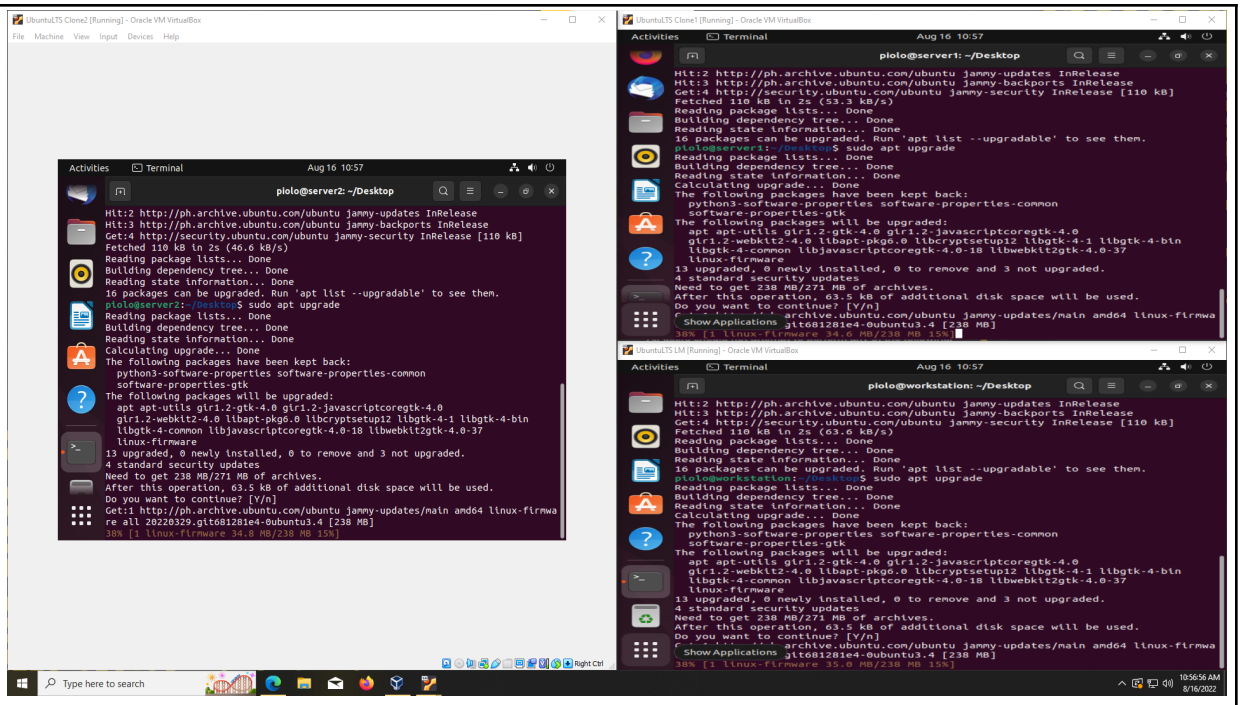
```
127.0.0.1       localhost
127.0.1.1       workstation

# 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
```

After Rebooting



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.



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

Server 1

```
piolo@server1:~/Desktop$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 openssh-sftp-server
amd64 1:8.9p1-3 [38.8 kB]
```

Server 2

```
piolo@server2:~/Desktop$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Local Machine

```
piolo@workstation:~/Desktop$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 openssh-sftp-server
amd64 1:8.9p1-3 [38.8 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 openssh-server amd64
 1:8.9p1-3 [434 kB]
10% [2 openssh-server 11.1 kB/434 kB 3%]
```

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

```
piolo@server1:~/Desktop$ sudo service ssh start
^[[A^[[Apiolo@server1:~/Desktop$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Tue 2022-08-16 11:39:10 PST; 56s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 15020 (sshd)
      Tasks: 1 (limit: 2896)
     Memory: 1.7M
        CPU: 29ms
     CGroup: /system.slice/ssh.service
             └─15020 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 16 11:39:10 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 16 11:39:10 server1 sshd[15020]: Server listening on 0.0.0.0 port 22.
Aug 16 11:39:10 server1 sshd[15020]: Server listening on :: port 22.
Aug 16 11:39:10 server1 systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

Server 2

```
piolo@server2:~/Desktop$ sudo service ssh start
piolo@server2:~/Desktop$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Tue 2022-08-16 11:53:28 PST; 54s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 14838 (sshd)
      Tasks: 1 (limit: 2896)
     Memory: 1.7M
        CPU: 26ms
     CGroup: /system.slice/ssh.service
             └─14838 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 16 11:53:28 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 16 11:53:28 server2 sshd[14838]: Server listening on 0.0.0.0 port 22.
Aug 16 11:53:28 server2 sshd[14838]: Server listening on :: port 22.
Aug 16 11:53:28 server2 systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

Local Machine

```
piolo@workstation:~/Desktop$ sudo service ssh start
piolo@workstation:~/Desktop$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Tue 2022-08-16 11:28:45 PST; 57s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 15087 (sshd)
      Tasks: 1 (limit: 2896)
     Memory: 1.7M
        CPU: 22ms
     CGroup: /system.slice/ssh.service
             └─15087 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 16 11:28:45 workstation systemd[1]: Starting OpenBSD Secure Shell server...
Aug 16 11:28:45 workstation sshd[15087]: Server listening on 0.0.0.0 port 22.
Aug 16 11:28:45 workstation sshd[15087]: Server listening on :: port 22.
Aug 16 11:28:45 workstation systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

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

```
piolo@server1:~/Desktop$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
piolo@server1:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
piolo@server1:~/Desktop$ sudo ufw status
Status: active

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

piolo@server1:~/Desktop$
```

Server 2

```
piolo@server2:~/Desktop$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
piolo@server2:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
piolo@server2:~/Desktop$ sudo ufw status
Status: active

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

piolo@server2:~/Desktop$
```

Local Machine

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

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

piolo@workstation:~/Desktop$
```

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: 192.168.56.103
   1.2 Server 2 IP address: 192.168.56.101
   1.3 Server 3 IP address: 192.168.56.102 (local machine)

Server 1

```
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::4543:6cd2:cf26:6a98  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ab:01:a8  txqueuelen 1000  (Ethernet)
        RX packets 47  bytes 6124 (6.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62  bytes 7870 (7.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Server 2

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.2
        inet6 fe80::b54:a48e:5c97:ee4b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0c:cc:50  txqueuelen 1000  (Ethernet)
        RX packets 88  bytes 12489 (12.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68  bytes 8370 (8.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Local Machine

```
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::d059:36ac:c9e0:b553  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:b7:f3:ae  txqueuelen 1000  (Ethernet)
        RX packets 68  bytes 9175 (9.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61  bytes 7996 (7.9 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

```
piolo@workstation:~/Desktop$ 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=0.578 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.468 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.402 ms
```

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

```
piolo@workstation:~/Desktop$ 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=0.559 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.400 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.348 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.446 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.233 ms
```

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

```
piolo@server1:~/Desktop$ 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=0.683 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.415 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.424 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.399 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

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.*

```
piolo@workstation:~/Desktop$ ssh piolo@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established
.
ED25519 key fingerprint is SHA256:5cBKdxwRfTlSR6kvrV5i+GNOayaWaUyQRoI3kPO3CX4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known host
s.
piolo@192.168.56.103's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

piolo@server1:~$
logout
Connection to 192.168.56.103 closed.
piolo@workstation:~/Desktop$
```

3. Do the same for Server 2.

```
piolo@workstation:~/Desktop$ ssh piolo@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established
.
ED25519 key fingerprint is SHA256:sO9fKPcopySDfKJR1tpts9ojvlVZeUvEsOsJUS/zkgg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known host
s.
piolo@192.168.56.101's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.

Last login: Tue Aug 16 12:09:54 2022 from 192.168.56.103
piolo@server2:~$
logout
Connection to 192.168.56.101 closed.
piolo@workstation:~/Desktop$
```

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)

```
127.0.0.1       localhost
127.0.1.1       workstation
192.168.56.103  server1
192.168.56.101  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
```

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.

Local Machine to Server 1

```
piolo@workstation:~/Desktop$ ssh piolo@server1
The authenticity of host 'server1 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:5cBKdxwRfTlSR6kvrV5i+GNOayaWaUyQRoI3kPO3CX4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server1' (ED25519) to the list of known hosts.
piolo@server1's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.

Last login: Tue Aug 16 12:08:51 2022 from 192.168.56.102
piolo@server1:~$
logout
Connection to server1 closed.
piolo@workstation:~/Desktop$
```

Local Machine to Server 2

```
piolo@workstation:~/Desktop$ ssh piolo@server2
The authenticity of host 'server2 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:sO9fKPcopySDfKJR1tpts9ojvlVZeUvEsOsJUS/zkgg.
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.
piolo@server2's password:
Permission denied, please try again.
piolo@server2's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.

Last login: Tue Aug 16 12:10:49 2022 from 192.168.56.102
piolo@server2:~$
```

Reflections:

Answer the following:

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

    - We are able to achieve the use of using the IP address in SSH commands by performing the 4 in task 4 which we are editing the /etc/hosts file where we add the IP address of server1 and server2 corresponding to its name. In this way we are stating an alias to an IP address where we can use it in accessing the SSH servers using only the username of the server.

2. How secured is SSH?

    - SSH uses the TCP protocol in port 22 in connecting to the remote machine but this connection is not secured by default. The SSH does is it first encrypts the packets that are being sent in the receiver machine then decrypts it after accepting the packet and vice versa. The other reason why SSH is connected is because it uses a multiplex multiple connect which creates a tunnel over the TCP connection. It also uses authentication when connecting to a machine that provides a high level of security but it can be brute forced sometimes.

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. "