Name:	Date Performed:
Course/Section:	Date Submitted:
Instructor:	Semester and SY:

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

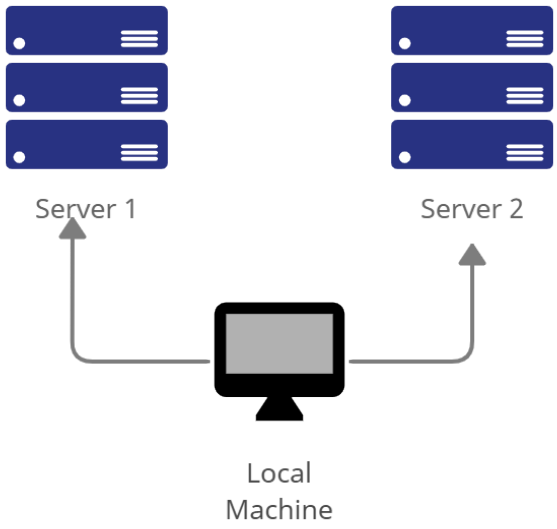
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

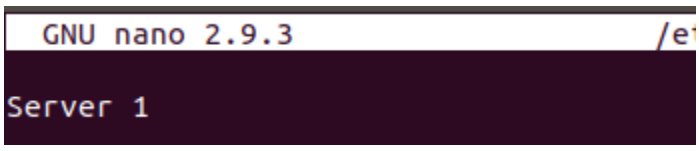
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



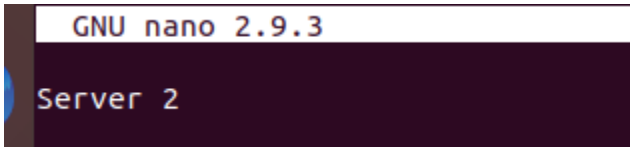
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

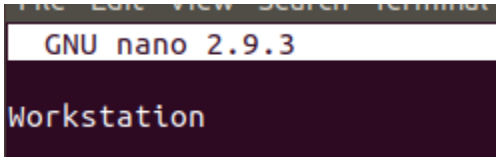
1.1 Use server1 for Server 1



1.2 Use server2 for Server 2

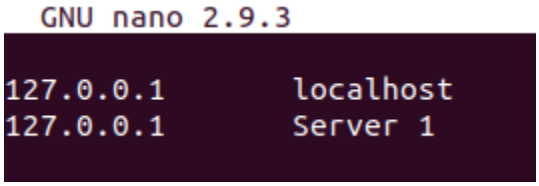


1.3 Use workstation for the Local Machine

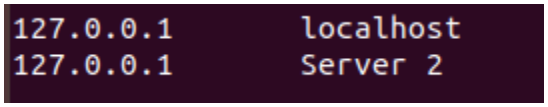


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

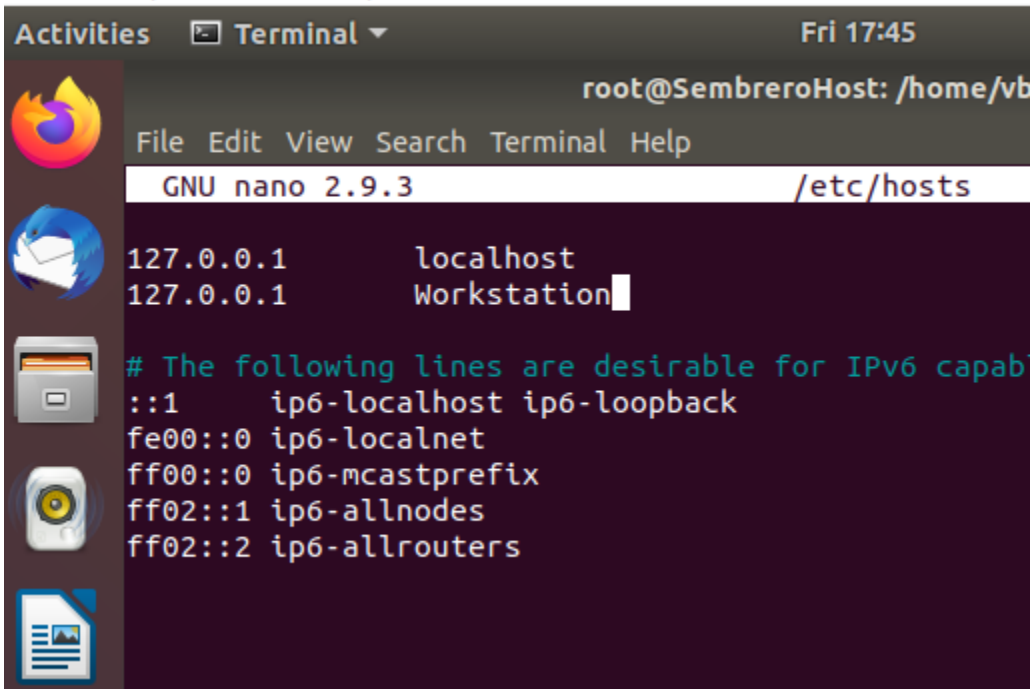
2.1 Type 127.0.0.1 server 1 for Server 1



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.

```
root@SembreroHost:/home/vboxuser# sudo apt update
```

```
root@SembreroHost:/home/vboxuser# 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*

```
root@SembreroHost:/home/vboxuser# sudo service ssh start
root@SembreroHost:/home/vboxuser# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
     Active: active (running) since Fri 2025-08-08 17:34:01 +08; 6min ago
   Main PID: 20245 (sshd)
      Tasks: 1 (limit: 2318)
     CGroup: /system.slice/ssh.service
             └─20245 /usr/sbin/sshd -D
```

4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*

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

   4.2 *sudo ufw enable*

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

   4.3 *sudo ufw status*

```
root@SembreroHost:/home/vboxuser# 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.**109**
   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

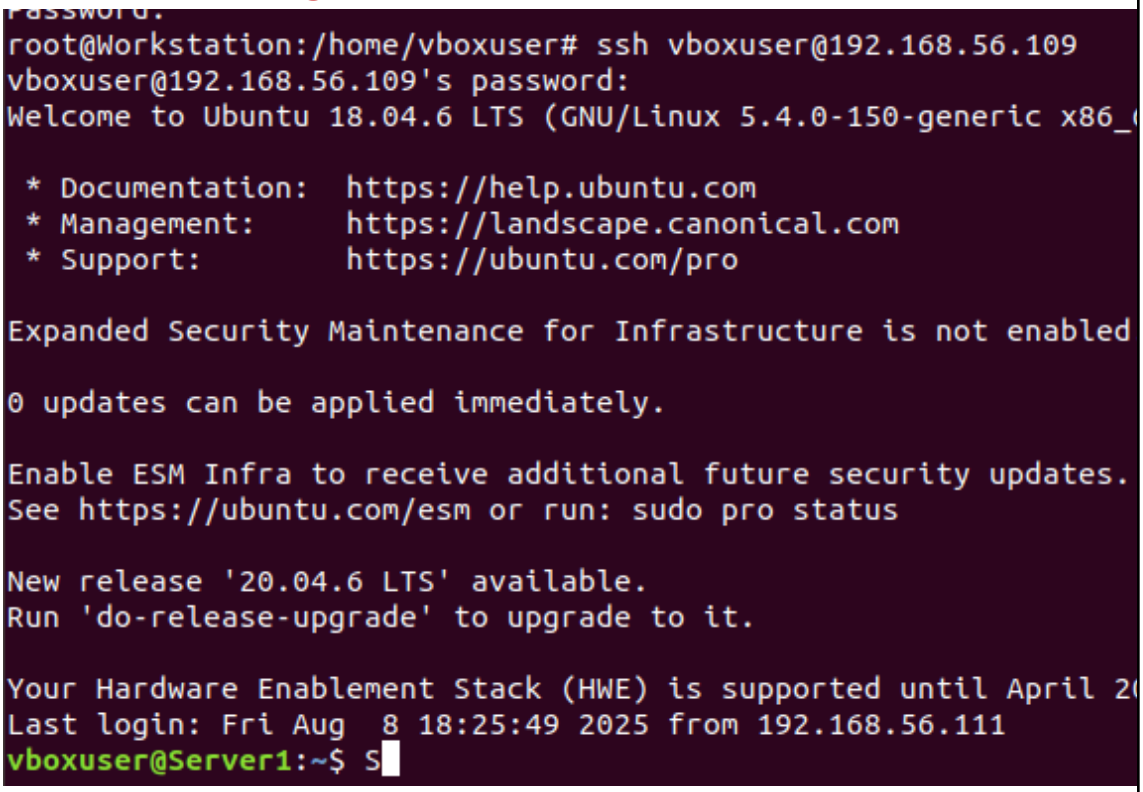
Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

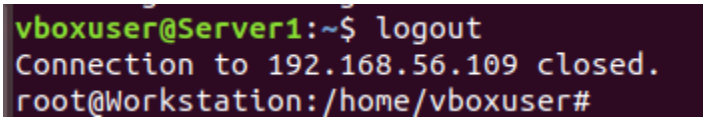
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

1.2 Enter the password for server 1 when prompted

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.

```
vboxuser@Server2:~$ ssh vboxuser@192.168.56.109
The authenticity of host '192.168.56.109 (192.168.56.109)' can't b
.
ECDSA key fingerprint is SHA256:tg4x/3aM6dZK4Fk985vvPD9M6F1+xZi2QL
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.56.109' (ECDSA) to the list of
vboxuser@192.168.56.109'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/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

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: Fri Aug  8 18:29:08 2025 from 192.168.56.111
vboxuser@Server1:~$
```

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)

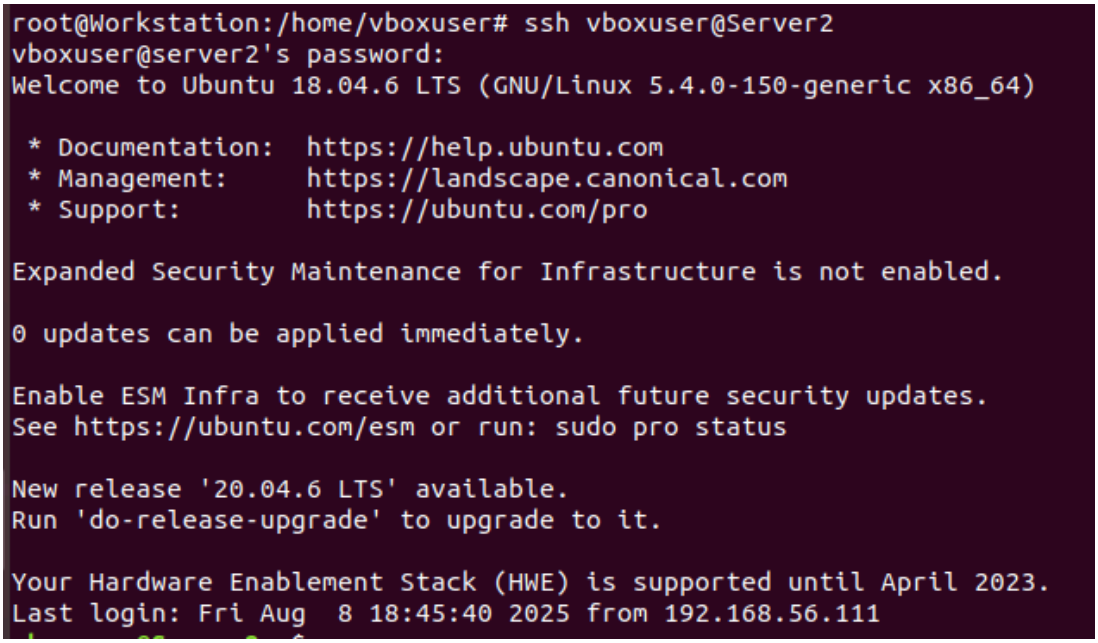
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

```
  GNU nano 2.9.3                        /etc/hosts

127.0.0.1       localhost
127.0.0.1       Workstation

# The following lines are desirable for IPv6 capable
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

Server 1 192.168.56.109
Server 2 192.168.56.110
```

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.

```
root@Workstation:/home/vboxuser# ssh vboxuser@Server1
vboxuser@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/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

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

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: Fri Aug  8 18:43:23 2025 from 192.168.56.111
```



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands? **Because we already include hostnames including their IP address inside nano /etc/hosts from the workstation server.**
2. How secured is SSH? **It is very secure because you'll need first to include their hostnames and IP address to remote or change into a server with just one machine.**