Name: Ramirez, Kiel Louis A.	**Date Performed: Aug 8, 2025**
Course/Section:CPE 212-CPE31S2	**Date Submitted:Aug 8, 2025**
Instructor: sir Robin	**Semester and SY: 2025-2026**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

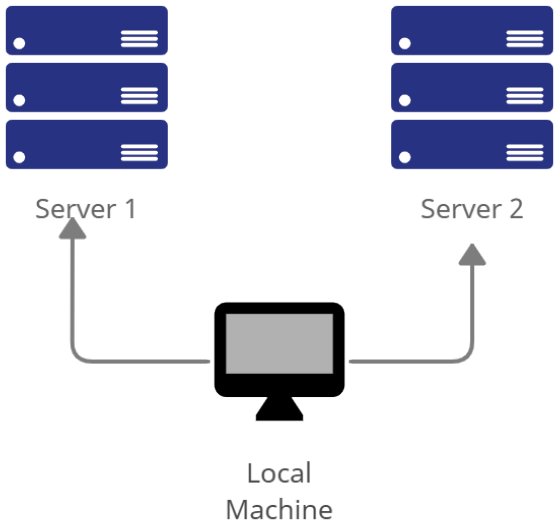
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

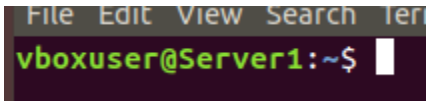
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



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

```
vboxuser@Server2:~$
```

1.3 Use workstation for the Local Machine

```
vboxuser@LocalMachine:~$
```

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       Server 1
```

2.2 Type 127.0.0.1 server 2 for Server 2

```
127.0.0.1       Server 2
```

2.3 Type 127.0.0.1 workstation for the Local Machine

```
127.0.0.1       Workstation
```

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.

```
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-150-generic
Found initrd image: /boot/initrd.img-5.4.0-150-generic
Found linux image: /boot/vmlinuz-4.18.0-15-generic
Found initrd image: /boot/initrd.img-4.18.0-15-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
Processing triggers for initramfs-tools (0.130ubuntu3.13) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for dbus (1.12.2-1ubuntu1.4) ...
root@LocalMachine:/home/vboxuser#
```

```
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-150-generic
Found initrd image: /boot/initrd.img-5.4.0-150-generic
Found linux image: /boot/vmlinuz-4.18.0-15-generic
Found initrd image: /boot/initrd.img-4.18.0-15-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
Processing triggers for initramfs-tools (0.130ubuntu3.13) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for dbus (1.12.2-1ubuntu1.4) ...
root@Server1:/home/vboxuser#
```

```
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-150-generic
Found initrd image: /boot/initrd.img-5.4.0-150-generic
Found linux image: /boot/vmlinuz-4.18.0-15-generic
Found initrd image: /boot/initrd.img-4.18.0-15-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
Processing triggers for initramfs-tools (0.130ubuntu3.13) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-150-generic
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for dbus (1.12.2-1ubuntu1.4) ...
root@Server2:/home/vboxuser#
```

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

```
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:V84qyFsUyS3Cfz15QzeMHxCwc8ZgJO+6jETpqCRq6KQ root@LocalMachine (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:S3NAdpRRE70+Q9zYO2QE/xwUkTgCQVTSTecFHJK2Gh8 root@LocalMachine (ECDSA
)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:jbLkQQMJfV2tLhfBJ0nTASjeFy7SFakRQQTr/+tUHkI root@LocalMachine (ED255
19)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.serv
ice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/
systemd/system/ssh.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.2) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.57) ...
root@LocalMachine:/home/vboxuser#
```

```
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:HlkVnPhzuvXDMwAeuU/M9QzO38KihjO2qdAVUl1QmL0 root@Server1 (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:GKYzLWfvcyHgr0F6c23w6Yzu6FMPp3NENh6KewojV5s root@Server1 (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:kPQioiozRqSI4dCy7laMxUDE6GdGbb//pttPXqwnZbg root@Server1 (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.serv
ice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/
systemd/system/ssh.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.2) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.57) ...
root@Server1:/home/vboxuser#
```

```
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:ivw0lI3XrqoSKkNJX6xS8YQRaBuCh2z4CV4Bse9udLw root@Server2 (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:zguxLYUA2OKxHcyjq/KeDbvnbzFRaDJ3XLfaCpXGxIw root@Server2 (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:7VVerHeC2rC4T+/SXwRP5YbzxoJ9TWkwbse1mzOOac8 root@Server2 (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.serv
ice.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/
systemd/system/ssh.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.2) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.57) ...
root@Server2:/home/vboxuser#
```

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

   3.1 *sudo service ssh start*

```
root@LocalMachine:/home/vboxuser# sudo service ssh start
```

```
root@Server1:/home/vboxuser# sudo service ssh start
```

```
root@Server2:/home/vboxuser# sudo service ssh start
```

   3.2 *sudo systemctl status ssh*

```
root@LocalMachine:/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 17:46:51 +08; 3min 50s ago
 Main PID: 19474 (sshd)
    Tasks: 1 (limit: 2318)
   CGroup: /system.slice/ssh.service
           └─19474 /usr/sbin/sshd -D

Aug 08 17:46:51 LocalMachine systemd[1]: Starting OpenBSD Secure Shell server..
Aug 08 17:46:51 LocalMachine sshd[19474]: Server listening on 0.0.0.0 port 22.
Aug 08 17:46:51 LocalMachine sshd[19474]: Server listening on :: port 22.
Aug 08 17:46:51 LocalMachine systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```

```
root@Server1:/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 17:47:25 +08; 5min ago
   Main PID: 20695 (sshd)
      Tasks: 1 (limit: 2318)
     CGroup: /system.slice/ssh.service
             └─20695 /usr/sbin/sshd -D

Aug 08 17:47:25 Server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 08 17:47:25 Server1 sshd[20695]: Server listening on 0.0.0.0 port 22.
Aug 08 17:47:25 Server1 sshd[20695]: Server listening on :: port 22.
Aug 08 17:47:25 Server1 systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```

```
root@Server2:/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 17:47:40 +08; 5min ago
   Main PID: 19969 (sshd)
      Tasks: 1 (limit: 2318)
     CGroup: /system.slice/ssh.service
             └─19969 /usr/sbin/sshd -D

Aug 08 17:47:40 Server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 08 17:47:40 Server2 sshd[19969]: Server listening on 0.0.0.0 port 22.
Aug 08 17:47:40 Server2 sshd[19969]: Server listening on :: port 22.
Aug 08 17:47:40 Server2 systemd[1]: Started OpenBSD Secure Shell server.
lines 1-12/12 (END)
```

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

   4.1 *sudo ufw allow ssh*

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

```
root@Server1:/home/vboxuser# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@Server1:/home/vboxuser#
```

```
root@Server2:/home/vboxuser# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@Server2:/home/vboxuser#
```

   4.2 *sudo ufw enable*

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

```
root@Server1:/home/vboxuser# sudo ufw enable
Firewall is active and enabled on system startup
root@Server1:/home/vboxuser#
```

```
root@Server2:/home/vboxuser# sudo ufw enable
Firewall is active and enabled on system startup
root@Server2:/home/vboxuser#
```

4.3 *sudo ufw status*

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

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

root@LocalMachine:/home/vboxuser#
```

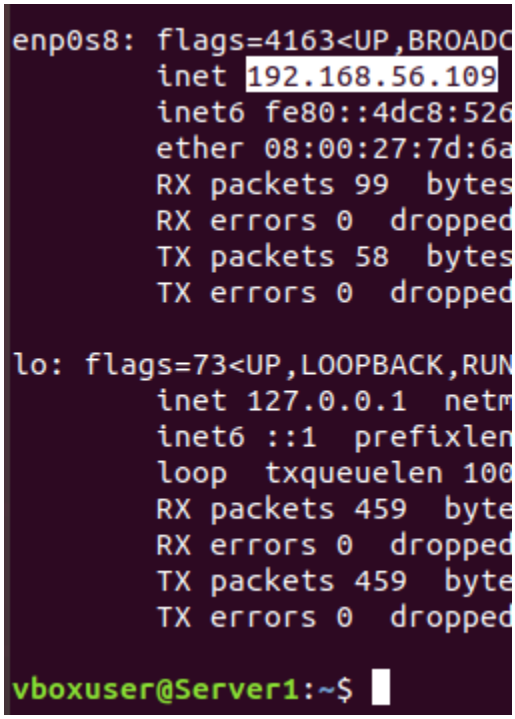
```
root@Server1:/home/vboxuser# sudo ufw enable
Firewall is active and enabled on system startup
root@Server1:/home/vboxuser# sudo ufw status
Status: active

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

root@Server1:/home/vboxuser#
```

```
root@Server2:/home/vboxuser# sudo ufw status
Status: active

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

root@Server2:/home/vboxuser#
```

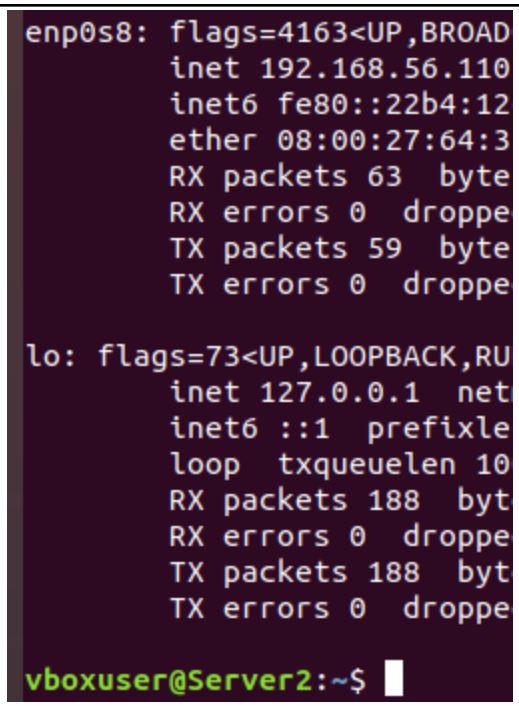
Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

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.

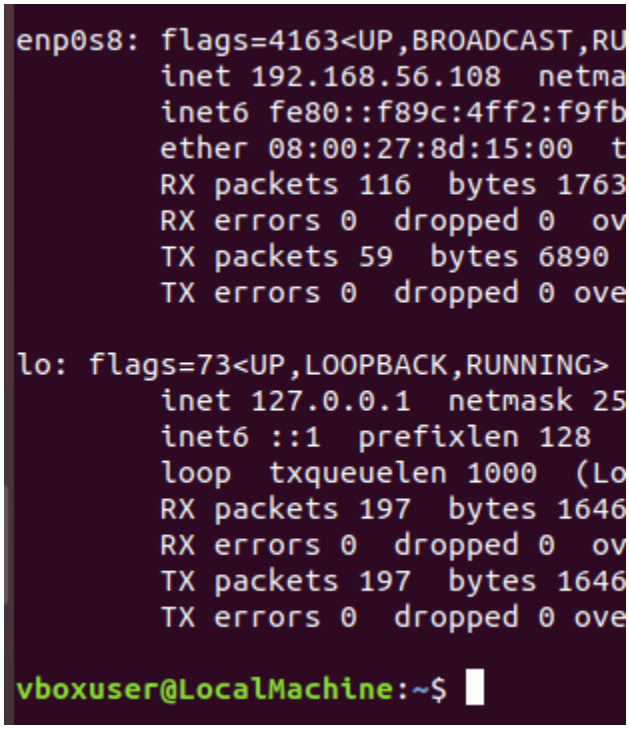
1.1 Server 1 IP address: 192.168.56.109



1.2 Server 2 IP address: 192.168.56.110



1.3 Server 3 IP address: 192.168.56.108



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 ✔

```
vboxuser@LocalMachine:~$ ping -c 4 192.168.56.110
PING 192.168.56.110 (192.168.56.110) 56(84) bytes of data.
64 bytes from 192.168.56.110: icmp_seq=1 ttl=64 time=0.884 ms
64 bytes from 192.168.56.110: icmp_seq=2 ttl=64 time=0.528 ms
64 bytes from 192.168.56.110: icmp_seq=3 ttl=64 time=0.404 ms
64 bytes from 192.168.56.110: icmp_seq=4 ttl=64 time=0.435 ms

--- 192.168.56.110 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056
rtt min/avg/max/mdev = 0.404/0.562/0.884/0.193 ms
```

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

```
vboxuser@Server1:~$ ping -c 4 192.168.56.110
PING 192.168.56.110 (192.168.56.110) 56(84) bytes of data.
64 bytes from 192.168.56.110: icmp_seq=1 ttl=64 time=0.912 m
64 bytes from 192.168.56.110: icmp_seq=2 ttl=64 time=0.425 m
64 bytes from 192.168.56.110: icmp_seq=3 ttl=64 time=0.451 m
64 bytes from 192.168.56.110: icmp_seq=4 ttl=64 time=0.916 m

--- 192.168.56.110 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056
rtt min/avg/max/mdev = 0.425/0.676/0.916/0.238 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*

```
vboxuser@LocalMachine:~$ ssh vboxuser@192.168.56.109
The authenticity of host '192.168.56.109 (192.168.56.109)' can't be established
.
ECDSA key fingerprint is SHA256:GKYzLWfvcyHgr0F6c23w6Yzu6FMPp3NENh6KewojV5s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.109' (ECDSA) to the list of known hosts.
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.
```

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*

```
vboxuser@Server1:~$
```

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

```
vboxuser@Server1:~$ logout
Connection to 192.168.56.109 closed.
vboxuser@LocalMachine:~$
```

3. Do the same for Server 2.

```
vboxuser@Server2:~$
```

```
vboxuser@Server2:~$ logout
Connection to 192.168.56.110 closed.
```

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       Local Machine
127.0.0.1       Workstation
192.168.56.109  server 1
192.168.56.110  server 2
```

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.

```
vboxuser@Server1:~$
```

```
vboxuser@Server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   -because of the connection of the ip address to connect the servers in the local machine
2. How secured is SSH?

-it is secured because lots of confirmation are needed and the ssh needs the ip address of the other server for connecting.