Name: Go, Meljune Royette G.	Date Performed: 8/18/2022
Course/Section: CPE31S23	Date Submitted: 8/18/2022
Instructor: Engr. Jonathan V. Taylar	Semester and SY: 1st Semester – 2022-2023

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

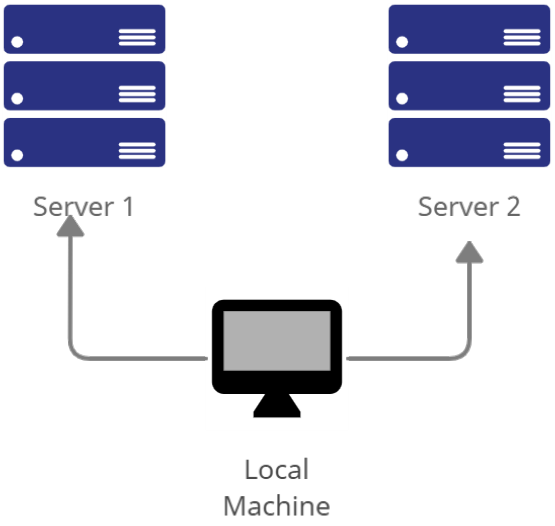
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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*).S



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

```
 GNU nano 6.2                                /etc/hostname *
server2
```

1.3 Use workstation for the Local Machine

```
 GNU nano 6.2                                /etc/hostname *
workstation
```

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.

```
junemel@junemel-VirtualBox:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
17 packages can be upgraded. Run 'apt list --upgradable' to see them.
junemel@junemel-VirtualBox:~$
```

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

```
junemel@junemel-VirtualBox:~$ 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 17 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] Y
```

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*

```
junemel@junemel-VirtualBox:~$ sudo service ssh start
junemel@junemel-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Thu 2022-08-18 10:32:38 PST; 1min 11s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3008 (sshd)
      Tasks: 1 (limit: 2288)
     Memory: 1.7M
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─3008 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 18 10:32:38 junemel-VirtualBox systemd[1]: Starting OpenBSD Secure Shell se>
Aug 18 10:32:38 junemel-VirtualBox sshd[3008]: Server listening on 0.0.0.0 port>
Aug 18 10:32:38 junemel-VirtualBox sshd[3008]: Server listening on :: port 22.
Aug 18 10:32:38 junemel-VirtualBox systemd[1]: Started OpenBSD Secure Shell ser>
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*

```
junemel@junemel-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
junemel@junemel-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
junemel@junemel-VirtualBox:~$ 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: 192.168.56.107
   1.2 Server 2 IP address: 192.168.56.106
   1.3 Server 3 IP address: 192.168.56.105
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*

```
$ ssh junemel@192.168.56.107
junemel@192.168.56.107'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

17 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


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.
```

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

```
junemel@server1:~$
logout
Connection to 192.168.56.107 closed.
```

3. Do the same for Server 2.

```
$ ssh junemel@192.168.56.106
The authenticity of host '192.168.56.106 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:k9CNOlA9zBOkOUwx8ZA1Y+VEIbBdcLDKO5AQWdqvKmk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.106' (ED25519) to the list of known hosts
.
junemel@192.168.56.106'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

17 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


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.
```

```
junemel@server2:~$
logout
Connection to 192.168.56.106 closed.
```

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

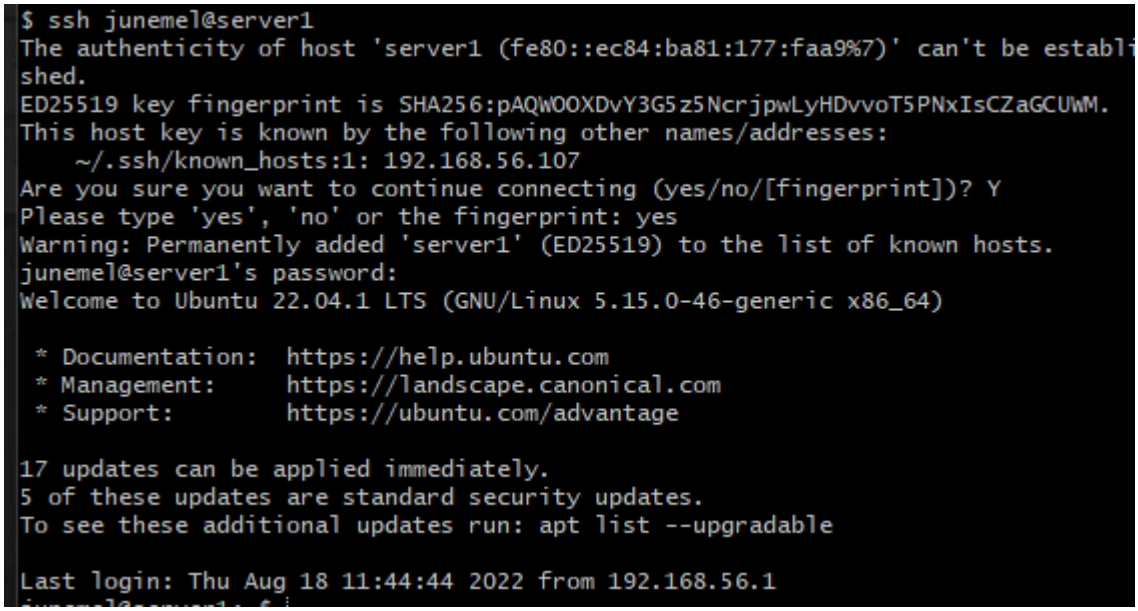
4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)

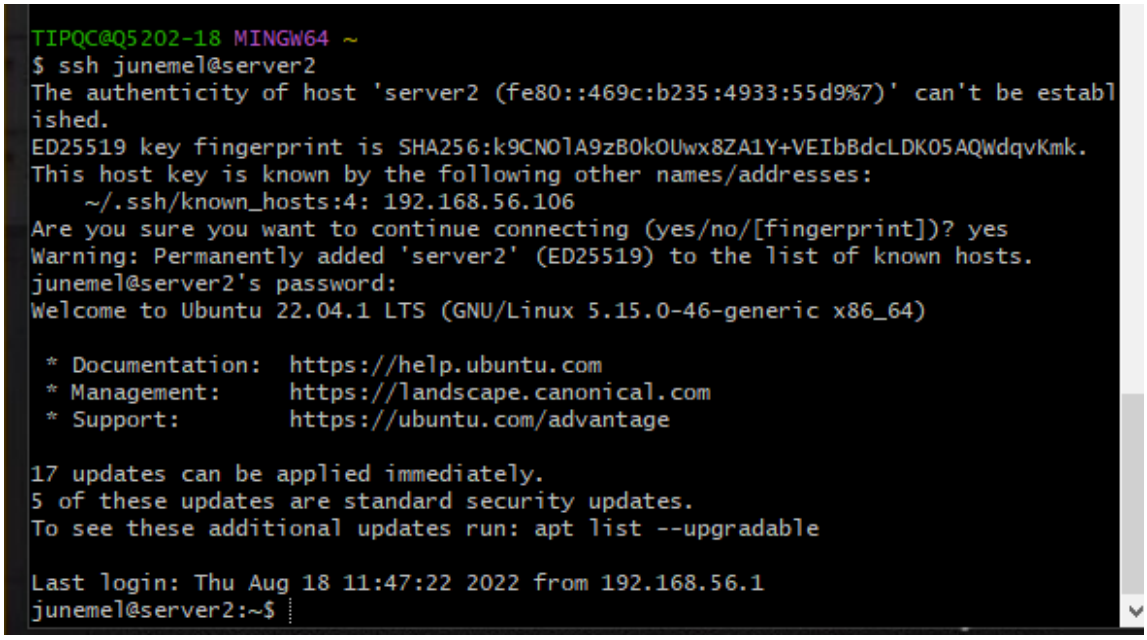
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

```
  GNU nano 6.2                          /etc/hosts *
127.0.0.1       localhost
127.0.0.1       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

192.168.56.107 server 1
192.168.56.106 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.





Reflections:

Answer the following:

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

   Hostnames was used because we edited the contents of the etc/hostnames. We Added the server 1 and server 2 and also their respective IP Addresses.

2. How secured is SSH?

   SSH is secured because it includes Hashing algorithms such as SHA-2**.** The commands of SSH is executed remotely. SSH uses the client-server model, connecting a Secure Shell client application, which is the end where the session is displayed, with an SSH server, which is the end where the session runs. SSH implementations often include support for application protocols used for terminal emulation or file transfers.