|	
Name: Bencito, Sonny Jay	Date Performed: 8/20/2022
Course/Section: CPE32S24	Date Submitted: 8/25/2022
Instructor: Engr. Jonathan Taylar	Semester and SY: 1st sem - 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*).

Server 1   Server 2
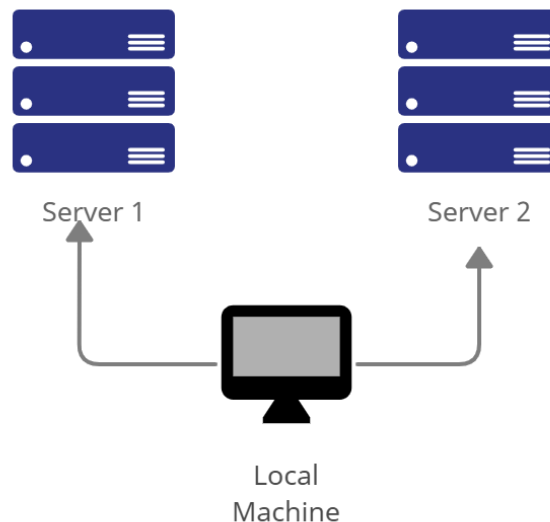
Local
Machine

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

```
bencito@bencito-VirtualBox:~$ sudo nano /etc/hostname
```

bencito@bencito-VirtualBox: ~

```
  GNU nano 6.2                    /etc/hostname *
Server 1
```

1.2 Use server2 for Server 2


```
                            bencito@bencito-VirtualBox: ~
  GNU nano 6.2                        /etc/hostname
Server 2
```

1.3 Use workstation for the Local Machine


```
                            bencito@bencito-VirtualBox: ~
  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


```
bencito@bencito-VirtualBox:~$ sudo nano /etc/hosts

                            bencito@bencito-VirtualBox: ~
  GNU nano 6.2                        /etc/hosts
127.0.0.1        localhost
127.0.1.1        Server 1
```

2.2 Type 127.0.0.1 server 2 for Server 2


```
                            bencito@bencito-VirtualBox: ~
  GNU nano 6.2                        /etc/hosts *
127.0.0.1        localhost
127.0.1.1        Server 2
```

2.3 Type 127.0.0.1 workstation for the Local Machine


```
                            bencito@bencito-VirtualBox: ~
  GNU nano 6.2                        /etc/hosts *
127.0.0.1        localhost
127.0.1.1        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.


```
bencito@bencito-VirtualBox:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for bencito:
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
All packages are up to date.
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.
bencito@bencito-VirtualBox:~$
```

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

```
bencito@bencito-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 0 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
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]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ncurses-term all 6.3
-2 [267 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.
11-0ubuntu1 [10.1 kB]
Fetched 751 kB in 2s (424 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 196178 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3) ...
Selecting previously unselected package openssh-server.
```

```
  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] y
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]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ncurses-term all 6.3
-2 [267 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.
11-0ubuntu1 [10.1 kB]
Fetched 751 kB in 0s (2,561 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 196179 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.9p1-3_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.3-2_all.deb ...
Unpacking ncurses-term (6.3-2) ...
Progress: [ 29%] [###############.........................................]
```

3. Verify if the SSH service has started by issuing the following commands:
    3.1 *sudo service ssh start*

```
bencito@bencito-VirtualBox:~$ sudo service ssh start
bencito@bencito-VirtualBox:~$
```

    3.2 *sudo systemctl status ssh*

```
bencito@bencito-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Sat 2022-08-20 09:57:05 PST; 2min 27s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 17908 (sshd)
      Tasks: 1 (limit: 2099)
     Memory: 1.7M
        CPU: 28ms
     CGroup: /system.slice/ssh.service
             └─17908 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 20 09:57:05 bencito-VirtualBox systemd[1]: Starting OpenBSD Secure Shell s>
Aug 20 09:57:05 bencito-VirtualBox sshd[17908]: Server listening on 0.0.0.0 po>
Aug 20 09:57:05 bencito-VirtualBox sshd[17908]: Server listening on :: port 22.
Aug 20 09:57:05 bencito-VirtualBox systemd[1]: Started OpenBSD Secure Shell se>
...skipping...
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Sat 2022-08-20 09:57:05 PST; 2min 27s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 17908 (sshd)
      Tasks: 1 (limit: 2099)
     Memory: 1.7M
        CPU: 28ms
     CGroup: /system.slice/ssh.service
             └─17908 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

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

   4.1 *sudo ufw allow ssh*

```
bencito@bencito-VirtualBox:~$ sudo ufw allow ssh
[sudo] password for bencito:
Rules updated
Rules updated (v6)
```

   4.2 *sudo ufw enable*

```
bencito@bencito-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

   4.3 *sudo ufw status*

```
bencito@bencito-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.

```
bencito@bencito-VirtualBox:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60
+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 0s (899 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 199055 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb .
..
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
```

1.1 Server 1 IP address: 192.168.56.<u>101</u>

```
inet 192.168.56.101
```

1.2 Server 2 IP address: 192.168.56.<u>102</u> This is Local Machine

```
inet 192.168.56.102
```

1.3 Server 3 IP address: 192.168.56.<u>103</u> This is Server 2

```
inet 192.168.56.103
```

2. Make sure that they can ping each other.

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

```
bencito@workstation:~$ ping -c 5 192.168.56.101
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=0.767 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.412 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.834 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.424 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.435 ms

--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.412/0.574/0.834/0.185 ms
```

Server 1 to Local Machine

```
bencito@bencito-VirtualBox:~$ ping -c 5 192.168.56.102
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=0.422 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.758 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.727 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.470 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.619 ms

--- 192.168.56.102 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4080ms
rtt min/avg/max/mdev = 0.422/0.599/0.758/0.134 ms
```

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

```
bencito@workstation:~$ ping -c 5 192.168.56.103
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=0.465 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.723 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.693 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.433 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.733 ms

--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.433/0.609/0.733/0.132 ms
```

Server 2 to Local Machine

```
bencito@Server2:~$ ping -c 5 192.168.56.102
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=0.441 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.492 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.722 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.767 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.726 ms

--- 192.168.56.102 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4088ms
rtt min/avg/max/mdev = 0.441/0.629/0.767/0.135 ms
```

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

```
bencito@bencito-VirtualBox:~$ ping -c 5 192.168.56.103
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=0.832 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.686 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.728 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.722 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.726 ms

--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4102ms
rtt min/avg/max/mdev = 0.686/0.738/0.832/0.049 ms
```

Server 2 to Server 1

```
bencito@bencito-VirtualBox:~$ ping -c 5 192.168.56.101
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=0.413 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.713 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.724 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.726 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.723 ms

--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4088ms
rtt min/avg/max/mdev = 0.413/0.659/0.726/0.123 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.*

3. Do the same for Server 2.

```
bencito@Server2:~$
logout
Connection to 192.168.56.103 closed.
```

```
bencito@Server2:~$ ssh bencito@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:pFlyJOAG4iwOCRjRElSTBPFODRN7QLQR8PlAwU2C8y4.
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.
bencito@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.
```

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

```
bencito@workstation:~$ sudo nano /etc/hosts
[sudo] password for bencito:
```

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.

```
                              bencito@workstation: ~                    Q  ≡  —  ◻  ✕

  GNU nano 6.2                          /etc/hosts *
127.0.0.1       localhost
127.0.1.1       workstation
192.168.56.101  Server 1
192.168.56.103  Server 2
```

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.

```
bencito@workstation:~$ ssh bencito@Server1
The authenticity of host 'server1 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:jkkT3ayyRxDdOBKDLTdniVcl1FYCtaBC7wG2K7OrXU8.
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.
bencito@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: Wed Aug 24 23:24:55 2022 from 192.168.56.101
```

Server 2

```
bencito@workstation:~$ ssh bencito@Server2
The authenticity of host 'server2 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:pFlyJOAG4iwOCRjRElSTBPFODRN7QLQR8PlAwU2C8y4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2' (ED25519) to the list of known hosts.
bencito@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: Wed Aug 24 23:29:27 2022 from 192.168.56.103
```

Reflections:

Answer the following:

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

   We are able to use the hostname by editing the local machine (main hosts) adding the two servers that we created and you need to set there IP address to use the hostname.

2. How secured is SSH?

   SSH server is have a strong encryption. While I'm doing the activity, I need to open the created server first before I can connect in on the main hosts. You need to have remember your password and they will give you a private key that only the owner knows it. They will ask you if you want to connect and type your password to make sure you are the owner. It is not easy to hack it because the system gives you a private key that only you can connect.

I affirm that I shall not give or receive any unauthorized help on this assignment and that all work shall be my own.