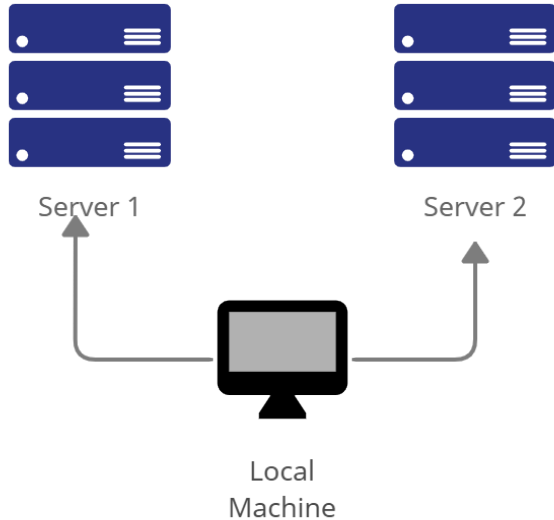


<b>Name: Escosia, Jerico</b>	<b>Date Performed: 08/30/24</b>
<b>Course/Section: CPE212-CPE32S21</b>	<b>Date Submitted: 08/30/24</b>
<b>Instructor: Sir</b>	<b>Semester and SY:</b>
<b>Activity 1: Configure Network using Virtual Machines</b>	
<b>1. Objectives:</b> 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 1.2. Set-up a Virtual Network and Test Connectivity of VMs	
<b>2. Discussion:</b>  <b>Network Topology:</b> Assume that you have created the following network topology in Virtual Machines, <i>provide screenshots for each task</i> . (Note: <i>it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine</i> ).	
 <pre> graph TD     LM[Local Machine] --- S1[Server 1]     LM --- S2[Server 2]     subgraph S1_Rack [Server 1]         S1_U1[ ]         S1_U2[ ]         S1_U3[ ]     end     subgraph S2_Rack [Server 2]         S2_U1[ ]         S2_U2[ ]         S2_U3[ ]     end       </pre>	
<b>Task 1:</b> 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. <ol style="list-style-type: none"> <li>Change the hostname using the command <i>sudo nano /etc/hostname</i> <ol style="list-style-type: none"> <li>Use server1 for Server 1</li> <li>Use server2 for Server 2</li> <li>Use workstation for the Local Machine</li> </ol> </li> <li>Edit the hosts using the command <i>sudo nano /etc/hosts</i>. Edit the second line.</li> </ol>	

## 2.1 Type 127.0.0.1 server 1 for Server 1

```
GNU nano 2.9.3 /etc/hosts

127.0.0.1    localhost
127.0.0.1    Escosiadesktop

# 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

```
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts Mod

127.0.0.1    localhost
127.0.0.1    Escosiadesktop

# 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

```
GNU nano 2.9.3 /etc/hosts

127.0.0.1    localhost
127.0.0.1    Escosiadesktop

# 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

```

```
root@Escosiadesktop:/home/jerico# hostnamectl
  Static hostname: Worskstation
  Transient hostname: Escosiadesktop
    Icon name: computer-vm
    Chassis: vm
  Machine ID: 93df0a5b33f94df89d9a38fc9c6d652f
  Boot ID: b9d01444aa5a4f5e810650f25cc9f5f3
  Virtualization: oracle
  Operating System: Ubuntu 18.04.2 LTS
    Kernel: Linux 4.18.0-15-generic
  Architecture: x86-64
root@Escosiadesktop:/home/jerico#
```

```
root@Escosiadesktop:/home/jerico# hostnamectl
  Static hostname: Server1
  Transient hostname: Escosiadesktop
    Icon name: computer-vm
    Chassis: vm
  Machine ID: 93df0a5b33f94df89d9a38fc9c6d652f
  Boot ID: 6d302f393f2c46c1ae5101f3e45cfbf5
  Virtualization: oracle
  Operating System: Ubuntu 18.04.2 LTS
    Kernel: Linux 4.18.0-15-generic
  Architecture: x86-64
```

```
root@Escosiadesktop:/home/jerico# sudo nano /etc/hostname
root@Escosiadesktop:/home/jerico# hostnamectl
  Static hostname: Server2
  Transient hostname: Escosiadesktop
    Icon name: computer-vm
    Chassis: vm
  Machine ID: 93df0a5b33f94df89d9a38fc9c6d652f
  Boot ID: 04479ca35da5489d9fcef7083d5fb47
  Virtualization: oracle
  Operating System: Ubuntu 18.04.2 LTS
    Kernel: Linux 4.18.0-15-generic
  Architecture: x86-64
root@Escosiadesktop:/home/jerico#
```

**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*.

```
root@Worskstation:/home/jerico# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
```

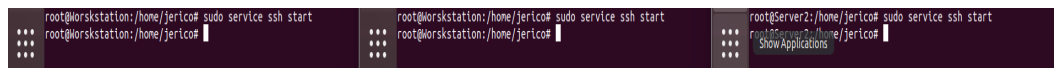
```

root@Worskstation:/home/jerico# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere rssh 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.

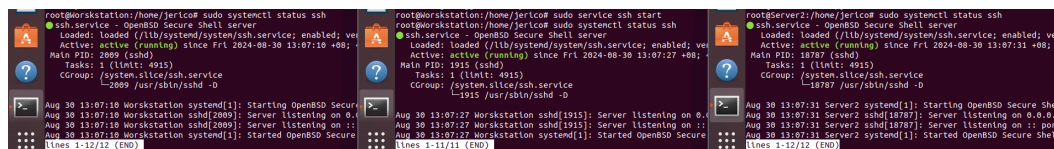
```

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

### 3.1 *sudo service ssh start*

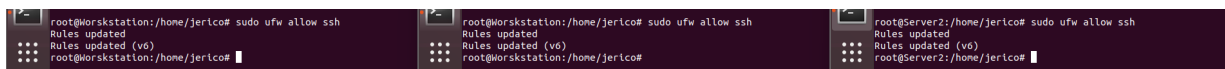


### 3.2 *sudo systemctl status ssh*

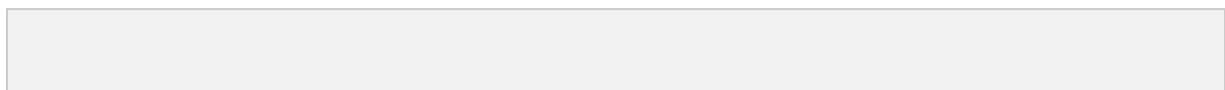


- Configure the firewall to all port 22 by issuing the following commands:

### 4.1 *sudo ufw allow ssh*



### 4.2 *sudo ufw enable*



### 4.3 *sudo ufw status*



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

- 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: 192.168.56.127

1.2 Server 2 IP address: 192.168.56.128

1.3 Server 3 IP address: 192.168.56.129

```

jerico@Worskstation:~$ ifconfig
enp83: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.127 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::157f:5a69:a3ee:de3b prefixlen 64 scopeid 0<link>
    ether 08:00:27:e5:cb:88 txqueuelen 1000 (Ethernet)
    RX packets 122 bytes 19916 (19.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 64 bytes 7478 (7.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0
    lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 359 bytes 26370 (26.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 359 bytes 26370 (26.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0

jerico@Worskstation:~$ ifconfig
enp83: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.128 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::6c0a:6a30:a5ac:fc10 prefixlen 64 scopeid 0<link>
    ether 08:00:27:07:ae:1d txqueuelen 1000 (Ethernet)
    RX packets 103 bytes 16241 (16.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 72 bytes 8573 (8.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0
    lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 359 bytes 26370 (26.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 359 bytes 26370 (26.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0

jerico@Server2:~$ ifconfig
enp83: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.129 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::157f:5a69:a3ee:de3b prefixlen 64 scopeid 0<link>
    ether 08:00:27:e5:cb:88 txqueuelen 1000 (Ethernet)
    RX packets 122 bytes 19916 (19.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 64 bytes 7478 (7.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0
    lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 359 bytes 26370 (26.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 359 bytes 26370 (26.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collision 0

```

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

```

jerico@Worskstation:~$ ping 192.168.56.128
PING 192.168.56.128 (192.168.56.128) 56(84) bytes of data.
64 bytes from 192.168.56.128: icmp_seq=1 ttl=64 time=0.899 ms
64 bytes from 192.168.56.128: icmp_seq=2 ttl=64 time=0.627 ms
64 bytes from 192.168.56.128: icmp_seq=3 ttl=64 time=0.451 ms
64 bytes from 192.168.56.128: icmp_seq=4 ttl=64 time=0.543 ms
64 bytes from 192.168.56.128: icmp_seq=5 ttl=64 time=0.459 ms
64 bytes from 192.168.56.128: icmp_seq=6 ttl=64 time=0.425 ms
64 bytes from 192.168.56.128: icmp_seq=7 ttl=64 time=0.503 ms
64 bytes from 192.168.56.128: icmp_seq=8 ttl=64 time=0.465 ms
64 bytes from 192.168.56.128: icmp_seq=9 ttl=64 time=0.455 ms
64 bytes from 192.168.56.128: icmp_seq=10 ttl=64 time=0.525 ms
64 bytes from 192.168.56.128: icmp_seq=11 ttl=64 time=0.333 ms
64 bytes from 192.168.56.128: icmp_seq=12 ttl=64 time=0.444 ms
64 bytes from 192.168.56.128: icmp_seq=13 ttl=64 time=0.453 ms
64 bytes from 192.168.56.128: icmp_seq=14 ttl=64 time=0.463 ms
64 bytes from 192.168.56.128: icmp_seq=15 ttl=64 time=0.489 ms
64 bytes from 192.168.56.128: icmp_seq=16 ttl=64 time=0.466 ms
^C
--- 192.168.56.128 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15391ms
rtt min/avg/max/mdev = 0.333/0.500/0.899/0.119 ms
jerico@Worskstation:~$ ping 192.168.56.129
PING 192.168.56.129 (192.168.56.129) 56(84) bytes of data.
64 bytes from 192.168.56.129: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.56.129: icmp_seq=2 ttl=64 time=0.482 ms
64 bytes from 192.168.56.129: icmp_seq=3 ttl=64 time=0.485 ms
64 bytes from 192.168.56.129: icmp_seq=4 ttl=64 time=0.582 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 jvtaylor@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, `jvtaylor@server1`

```
jerico@Worskstation:~$ ssh jerico@192.168.56.128
The authenticity of host '192.168.56.128 (192.168.56.128)' can't be established
.
ECDSA key fingerprint is SHA256:zofEiWARzMwXMnw6ZSqxuoQXQZgVIR/R1oGSiGSr0CI.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.56.128' (ECDSA) to the list of known hosts.
jerico@192.168.56.128'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

Your Hardware Enablement Stack (HWE) is supported until April 2023.

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
```

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

```
jerico@Worskstation:~$ logout
Connection to 192.168.56.128 closed.
```

3. Do the same for Server 2.

```
jerico@Worskstation:~$ ssh jerico@192.168.56.128
The authenticity of host '192.168.56.128 (192.168.56.128)' can't be established
.
ECDSA key fingerprint is SHA256:zofEiWARzMwXMnw6ZSqxuoQXQZgVIR/R1oGSiGSr0CI.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.56.128' (ECDSA) to the list of known hosts.
jerico@192.168.56.128'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

Your Hardware Enablement Stack (HWE) is supported until April 2023.

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
```

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.
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 jvtaylor@server1*. Enter the password when prompted. Verify that you

**Reflections:**

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
  - The machines was connected using DNS so we can connect using hostnames instead of IP address.
2. How secured is SSH?
  - SSH uses strong encryption that stores our logging info within the system and can be remotely access.