Name: Ballesteros, John Erwin S.	**Date Performed:** 8/22/2024
Course/Section: CpE31S2	**Date Submitted: 8/23/2024**
Instructor: Mr. Robin Valenzuela/ Engr. Maria Rizette Sayo	**Semester and SY:** 1st Semester '24-'25

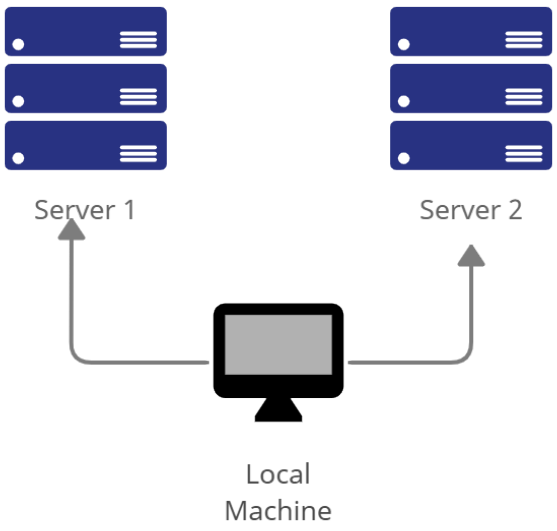
Activity 1: Configure Network using Virtual Machines

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*).

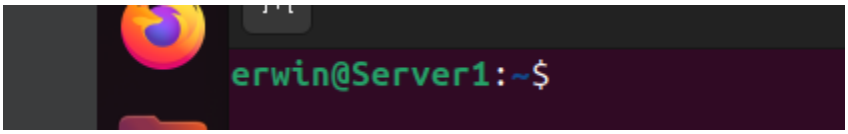


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

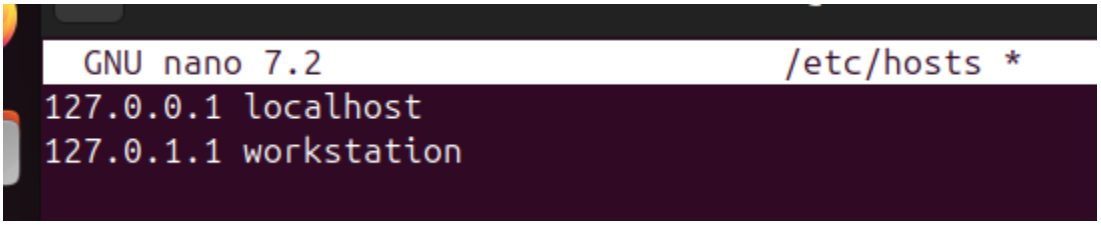
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.
   2.1 Type 127.0.0.1 server 1 for Server 1
   2.2 Type 127.0.0.1 server 2 for Server 2
   2.3 Type 127.0.0.1 workstation for the Local Machine

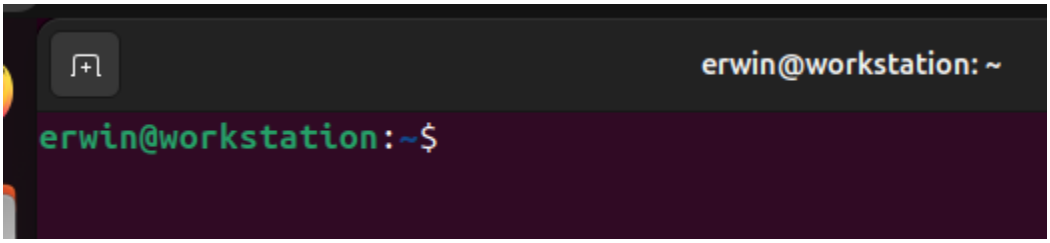
Server 1



Server 2

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.
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*
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*

Server 1

```
erwin@Server1:~$ sudo apt update
[sudo] password for erwin:
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

```
erwin@Server1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
erwin@Server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
erwin@Server1:~$ sudo service ssh start
erwin@Server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2024-08-23 15:15:33 PST; 5s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 13344 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 13346 (sshd)
      Tasks: 1 (limit: 9789)
     Memory: 1.2M (peak: 1.5M)
        CPU: 14ms
     CGroup: /system.slice/ssh.service
             └─13346 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

```
erwin@Server1:~$ sudo ufw allow ssh
[sudo] password for erwin:
Rules updated
Rules updated (v6)
erwin@Server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
erwin@Server1:~$ sudo ufw status
Status: active

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

Server 2

```
erwin@Server2:~$ sudo apt update
[sudo] password for erwin:
Warning: The unit file, source configuration file o
```

```
erwin@Server2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred d
```

```
erwin@Server2:~$ 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-client openssh-sftp-server ssh-import-i
```

```
erwin@Server2:~$ sudo service ssh start
erwin@Server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2024-08-23 15:49:18 PST; 5s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 14483 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 14485 (sshd)
      Tasks: 1 (limit: 9789)
     Memory: 1.2M (peak: 1.5M)
        CPU: 13ms
```

file://Server2/usr/lib/systemd/system/ssh.service

```
erwin@Server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
erwin@Server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
erwin@Server2:~$ sudo ufw status
Status: active

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

Workstation

```
erwin@workstation:~$ sudo apt update
[sudo] password for erwin:
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
```

```
erwin@workstation:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
erwin@workstation:~$ 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-client openssh-sftp-server ssh-i
```

```
erwin@workstation:~$ sudo service ssh start
erwin@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: active (running) since Fri 2024-08-23 14:56:26 PST; 10s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 13009 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 13011 (sshd)
      Tasks: 1 (limit: 9789)
     Memory: 1.2M (peak: 1.5M)
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             └─13011 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

```
erwin@workstation:~$ sudo ufw allow ssh
[sudo] password for erwin:
Rules updated
Rules updated (v6)
erwin@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
erwin@workstation:~$ sudo ufw status
Status: active

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

erwin@workstation:~$
```

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.___ **- 192.168.100.120**

```
erwin@Server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAS
        inet 192.168.100.120
```

1.2 Server 2 IP address: 192.168.56.___ **- 192.168.100.119**

```
erwin@Server2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCA
        inet 192.168.100.119
```

1.3 Server 3 IP address: 192.168.56.___ **- 192.168.100.118**

```
erwin@workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCA
        inet 192.168.100.118
```

2. Make sure that they can ping each other.

2.1 Connectivity test for Local Machine 1 to Server 1:

✅ Successful ☐ Not Successful

```
erwin@workstation:~$ ping 192.168.100.120 -c 5
PING 192.168.100.120 (192.168.100.120) 56(84) bytes of data.
64 bytes from 192.168.100.120: icmp_seq=1 ttl=64 time=0.430 ms
64 bytes from 192.168.100.120: icmp_seq=2 ttl=64 time=0.293 ms
64 bytes from 192.168.100.120: icmp_seq=3 ttl=64 time=0.276 ms
64 bytes from 192.168.100.120: icmp_seq=4 ttl=64 time=0.267 ms
64 bytes from 192.168.100.120: icmp_seq=5 ttl=64 time=0.296 ms

--- 192.168.100.120 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4051ms
rtt min/avg/max/mdev = 0.267/0.312/0.430/0.059 ms
```

2.2 Connectivity test for Local Machine 1 to Server 2:

✅ Successful ☐ Not Successful

```
erwin@workstation:~$ ping 192.168.100.119 -c 5
PING 192.168.100.119 (192.168.100.119) 56(84) bytes of data.
64 bytes from 192.168.100.119: icmp_seq=1 ttl=64 time=0.582 ms
64 bytes from 192.168.100.119: icmp_seq=2 ttl=64 time=0.317 ms
64 bytes from 192.168.100.119: icmp_seq=3 ttl=64 time=0.241 ms
64 bytes from 192.168.100.119: icmp_seq=4 ttl=64 time=0.392 ms
64 bytes from 192.168.100.119: icmp_seq=5 ttl=64 time=0.258 ms

--- 192.168.100.119 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4033ms
rtt min/avg/max/mdev = 0.241/0.358/0.582/0.123 ms
```

2.3 Connectivity test for Server 1 to Server 2:

✅ Successful ☐ Not Successful

```
erwin@Server1:~$ ping 192.168.100.119 -c 5
PING 192.168.100.119 (192.168.100.119) 56(84) bytes of data.
64 bytes from 192.168.100.119: icmp_seq=1 ttl=64 time=0.517 ms
64 bytes from 192.168.100.119: icmp_seq=2 ttl=64 time=0.278 ms
64 bytes from 192.168.100.119: icmp_seq=3 ttl=64 time=0.271 ms
64 bytes from 192.168.100.119: icmp_seq=4 ttl=64 time=0.228 ms
64 bytes from 192.168.100.119: icmp_seq=5 ttl=64 time=0.246 ms

--- 192.168.100.119 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
rtt min/avg/max/mdev = 0.228/0.308/0.517/0.106 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.
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 jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

Server 1

```
erwin@workstation:~$ ssh erwin@192.168.100.120
The authenticity of host '192.168.100.120 (192.168.100.120)' can't be establishe
d.
ED25519 key fingerprint is SHA256:p3/YJZpHJf8bvD0U5naNEaYScOkLWzDmqVAhLWvcULk.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.120' (ED25519) to the list of known host
s.
```

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

Last login: Fri Aug 23 17:55:03 2024 from 192.168.100.116
erwin@Server1:~$ echo hello world
hello world
erwin@Server1:~$
```

```
erwin@Server1:~$
logout
Connection to 192.168.100.120 closed.
erwin@workstation:~$
```

Server 2

```
erwin@workstation:~$ ssh erwin@192.168.100.119
The authenticity of host '192.168.100.119 (192.168.100.119)' can't be establis
d.
ED25519 key fingerprint is SHA256:p3/YJZpHJf8bvD0U5naNEaYScOkLWzDmqVAhLWvcULk.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

erwin@Server2:~$ echo hello world
hello world
erwin@Server2:~$
```

```
erwin@Server2:~$
logout
Connection to 192.168.100.119 closed.
erwin@workstation:~$
```

Sudo nano edit hosts

```
erwin@workstation: ~
  GNU nano 7.2                    /etc/hosts *
127.0.0.1 localhost
127.0.1.1 workstation
192.168.100.120 server1
192.168.100.119 server2

# The following lines are desirable for IPv6 capable hosts
```

Server 1

```
erwin@workstation:~$ ssh erwin@server1
The authenticity of host 'server1 (192.168.100.120)' can't be established.
ED25519 key fingerprint is SHA256:p3/YJZpHJf8bvD0U5naNEaYScOkLWzDmqVAhLWvcU
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
```

```
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug 23 18:21:58 2024 from 192.168.100.118
erwin@Server1:~$
```

Server 2

```
Connection to server1 closed.
erwin@workstation:~$ ssh erwin@server2
The authenticity of host 'server2 (192.168.100.119)' can't be established.
ED25519 key fingerprint is SHA256:p3/YJZpHJf8bvD0U5naNEaYScOkLWzDmqVAhLWvcULk.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
```

```
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug 23 18:24:06 2024 from 192.168.100.118
erwin@Server2:~$
```

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 of the server instead of the IP address is by defining the name of the device in the /etc/hosts.

2. **How secured is SSH?**

   SSH is secure because it is able to encrypt and authenticate the process by cryptography. It's usage is providing a secure method for server management and different specific tasks that require an encryption.