Name: Denzel Dinglasan	Date Performed: 17/08/2023
Course/Section:CPE 232 - CPE31S6	Date Submitted: 17/08/2023
Instructor: Dr. Jonathan Vidal Taylar	Semester and SY: 1st Sem 2023 - 2024

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

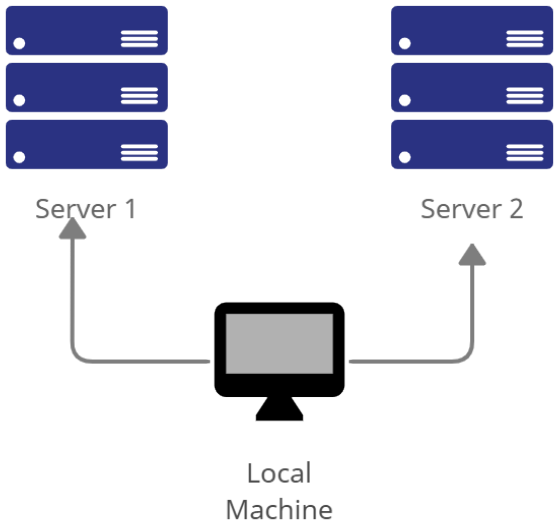
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

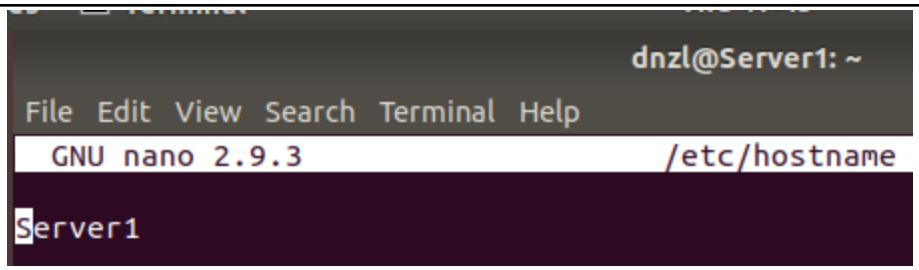
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



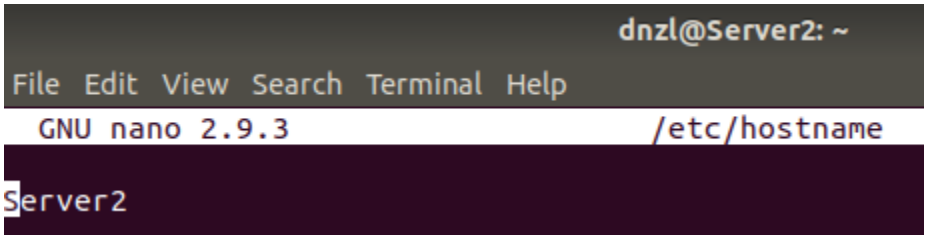
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

1. Change the hostname using the command *sudo nano /etc/hostname*

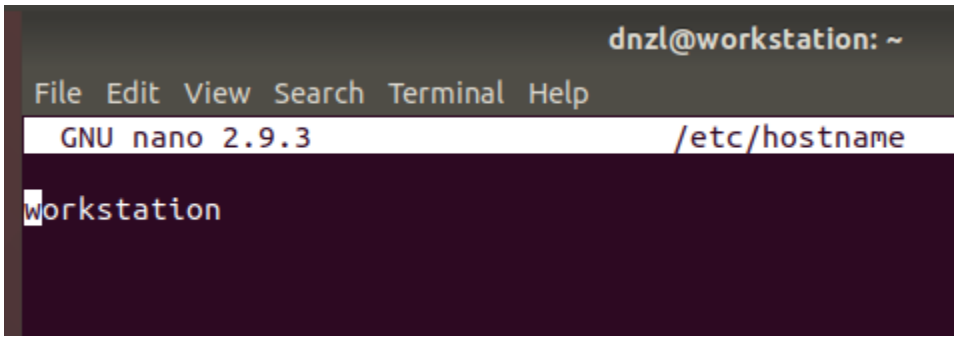
   1.1 Use server1 for Server 1



1.2 Use server2 for Server 2

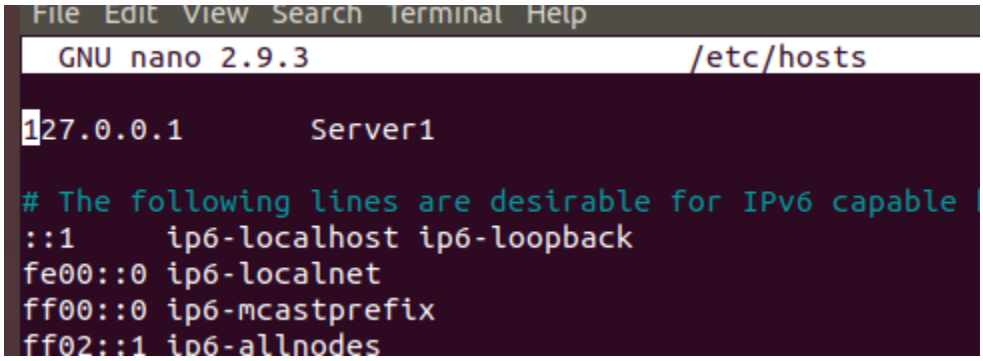


1.3 Use workstation for the Local Machine

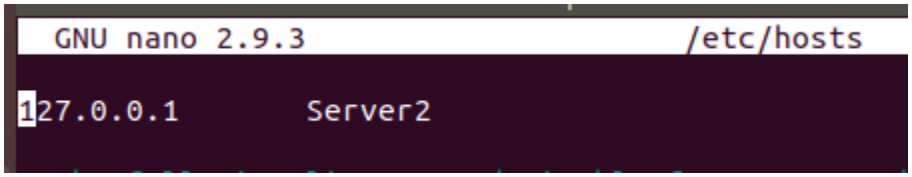


2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

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

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

127.0.0.1       workstation

# The following lines are desirable for IPv6 capab
```

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*

```
dnzl@workstation:~$ sudo service ssh start
dnzl@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pres
   Active: active (running) since Thu 2023-08-17 17:17:55 PST; 7min ago
 Main PID: 2073 (sshd)
    Tasks: 1 (limit: 4656)
   CGroup: /system.slice/ssh.service
           └─2073 /usr/sbin/sshd -D

Aug 17 17:17:55 workstation systemd[1]: Starting OpenBSD Secure Shell se
Aug 17 17:17:55 workstation sshd[2073]: Server listening on 0.0.0.0 port
Aug 17 17:17:55 workstation sshd[2073]: Server listening on :: port 22.
Aug 17 17:17:55 workstation systemd[1]: Started OpenBSD Secure Shell ser
```

```
dnzl@Server1:~$ sudo service ssh start
[sudo] password for dnzl:
dnzl@Server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pre
   Active: active (running) since Thu 2023-08-17 17:23:52 PST; 3min 24s
  Process: 1100 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status
  Process: 1099 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUC
  Process: 675 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SU
 Main PID: 685 (sshd)
    Tasks: 1 (limit: 4656)
   CGroup: /system.slice/ssh.service
           └─685 /usr/sbin/sshd -D

Aug 17 17:23:55 Server1 systemd[1]: Reloading OpenBSD Secure Shell serv
Aug 17 17:23:55 Server1 sshd[685]: Received SIGHUP; restarting.
Aug 17 17:23:55 Server1 systemd[1]: Reloaded OpenBSD Secure Shell serve
Aug 17 17:23:55 Server1 sshd[685]: Server listening on 0.0.0.0 port 22.
Aug 17 17:23:55 Server1 sshd[685]: Server listening on :: port 22.
```

```
dnzl@Server2:~$ sudo service ssh start
[sudo] password for dnzl:
dnzl@Server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor pr
     Active: active (running) since Thu 2023-08-17 17:24:27 PST; 3min 2
    Process: 1064 ExecReload=/bin/kill -HUP $MAINPID (code=exited, statu
    Process: 1063 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SU
    Process: 663 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/S
   Main PID: 675 (sshd)
      Tasks: 1 (limit: 4656)
     CGroup: /system.slice/ssh.service
             └─675 /usr/sbin/sshd -D

Aug 17 17:24:30 Server2 systemd[1]: Reloading OpenBSD Secure Shell ser
Aug 17 17:24:30 Server2 sshd[675]: Received SIGHUP; restarting.
Aug 17 17:24:30 Server2 systemd[1]: Reloaded OpenBSD Secure Shell serv
Aug 17 17:24:30 Server2 sshd[675]: Server listening on 0.0.0.0 port 22
```

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*

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

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

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

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

```
dnzl@Server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
dnzl@Server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
dnzl@Server2:~$ 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: 192.168.56.102
   1.2 Server 2 IP address: 192.168.56.103
   1.3 workstation IP address: 192.168.56.101
2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ☐ Successful ☐ Not Successful

```
dnzl@workstation:~$ ping 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=1.09 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.566 m
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.765 m
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.412 m
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.425 m
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.472 m
```

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

```
dnzl@workstation:~$ ping 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=0.957 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.487 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.860 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.961 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=1.06 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=1.54 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=1.09 ms
^Z
```

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

```
dnzl@Server1:~$ ping 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=1.18 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.406 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.445 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=0.453 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.427 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=0.463 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=64 time=0.415 ms
64 bytes from 192.168.56.103: icmp_seq=9 ttl=64 time=0.505 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*

```
dnzl@workstation:~$ ssh dnzl@192.168.56.102
```

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.*

```
dnzl@Server1:~$ logout
Connection to 192.168.56.102 closed.
dnzl@workstation:~$
```

2.1

3. Do the same for Server 2.

```
dnzl@workstation:~$ ssh dnzl@192.168.56.103
```

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

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 2.9.3                        /etc/hosts

127.0.0.1       workstation
192.168.56.102  Server1
192.168.56.103  Server2

# 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
```

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.

```
dnzl@workstation:~$ ssh dnzl@server1
The authenticity of host 'server1 (192.168.56.102)' can't be establishe
ECDSA key fingerprint is SHA256:Zt+9b/aBAy+qpt+nB3CE2ChCNioPYI8T6Cqud/k
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1' (ECDSA) to the list of known hosts.
dnzl@server1'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/advantage

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Thu Aug 17 17:54:07 2023 from 192.168.56.101
dnzl@Server1:~$
```

```
dnzl@workstation:~$ ssh dnzl@Server2
The authenticity of host 'server2 (192.168.56.103)' can't be establish
ECDSA key fingerprint is SHA256:vKKChS9DlLKESfwIO8XgxmChaTiMVrFKPxEgp8
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server2' (ECDSA) to the list of known host
dnzl@server2'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/advantage

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Thu Aug 17 17:39:35 2023 from 192.168.56.101
dnzl@Server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   By providing a host for the corresponding IP addresses.
2. How secured is SSH?

   SSH is typically regarded as a secure remote access technique. For optimal safety, use stronger passwords and other security measures.

Conclusions:

In this activity, I managed to learn how to create a virtual machine and connect to servers from my workstation.