| | |
|---|---|
| **Name: Gabiano, Chris Leonard A.** | **Date Performed: August 17, 2023** |
| **Course/Section: CPE232-S6** | **Date Submitted: August 17, 2023** |
| **Instructor: Dr. Jonathan Taylar** | **Semester and SY:** |

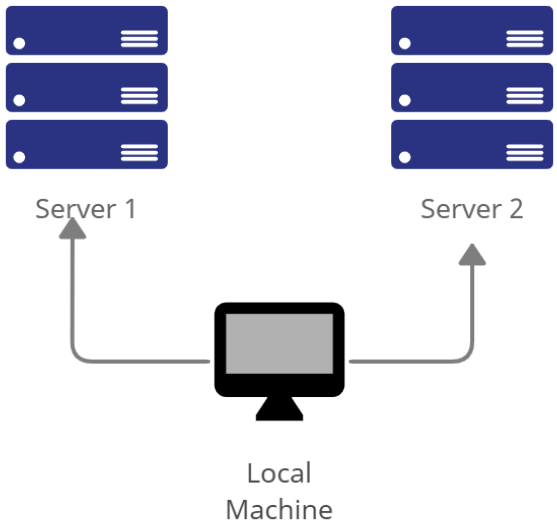
**Activity 1: Configure Network using Virtual Machines**

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:**

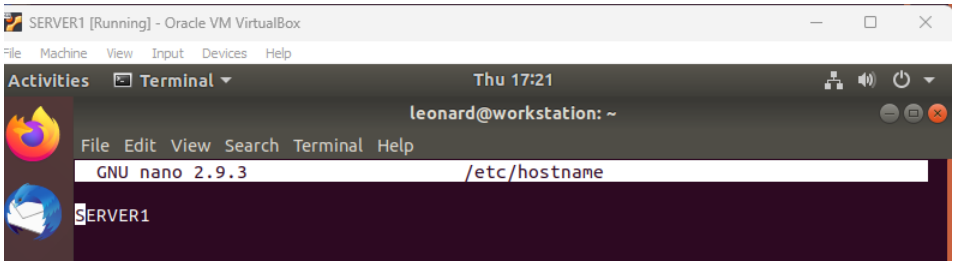
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



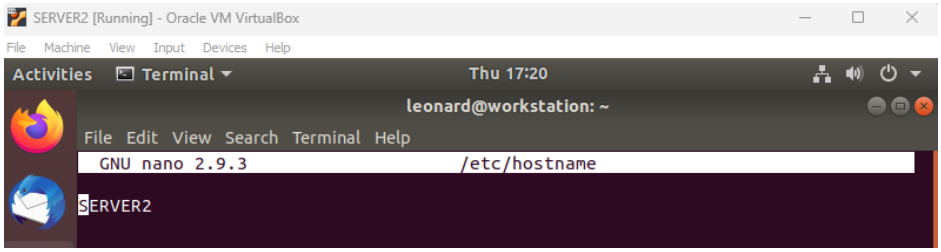

**Task 1**: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

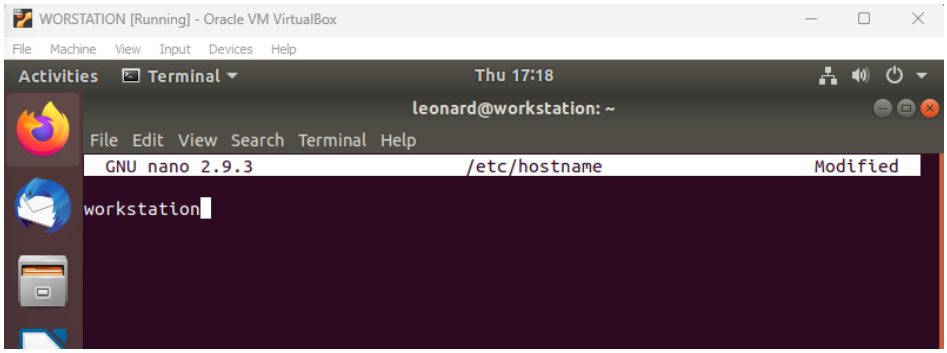
1.1 Use server1 for Server 1




1.2 Use server2 for Server 2

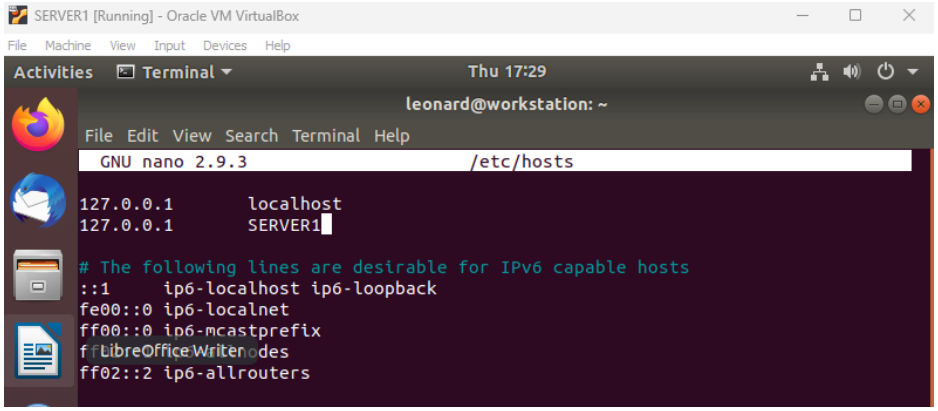



1.3 Use workstation for the Local Machine

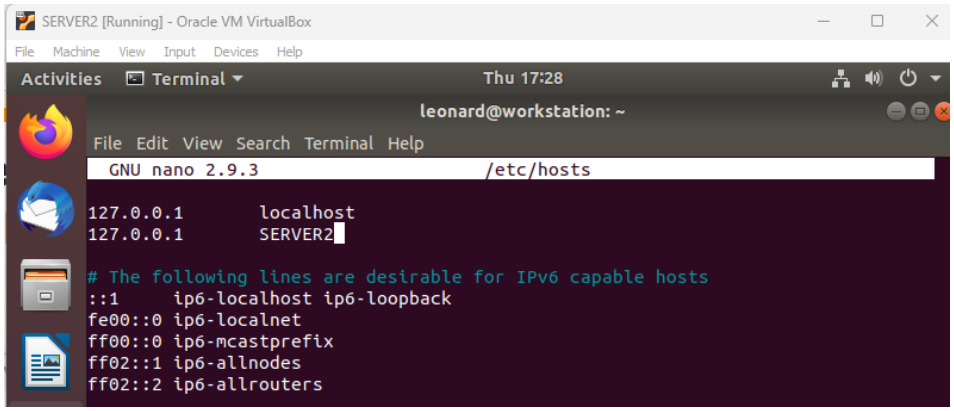



2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

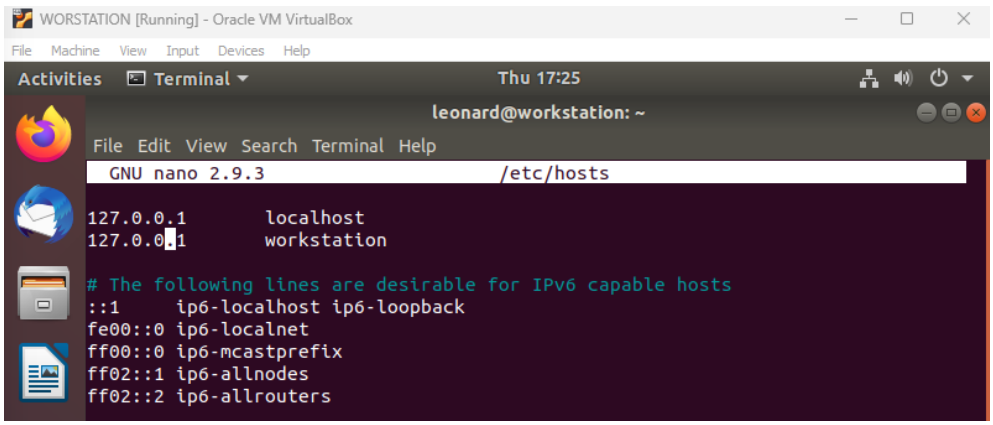
2.1 Type 127.0.0.1 server 1 for Server 1



2.2 Type 127.0.0.1 server 2 for Server 2




2.3 Type 127.0.0.1 workstation for the Local Machine




**Task 2**: Configure SSH on Servesr 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.

**workstation**

```
leonard@workstation:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
leonard@workstation:~$ sudo apt upgrade
```

**server1**

```
leonard@workstation:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
leonard@workstation:~$ sudo nano upgrade
leonard@workstation:~$ sudo apt upgrade
Reading package lists... Done
```

**server2**

```
leonard@workstation:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
leonard@workstation:~$ sudo apt upgrade
```

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

**workstation**

```
leonard@workstation:~$ 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.
```

**server1**

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
leonard@workstation:~$ 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
```

**server2**

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
leonard@workstation:~$ 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 requ
  libllvm7
```

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*

***workstation***

```
leonard@workstation:~$ sudo service ssh start
leonard@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-08-17 17:37:54 PST; 5min ago
 Main PID: 2653 (sshd)
    Tasks: 1 (limit: 4884)
   CGroup: /system.slice/ssh.service
           └─2653 /usr/sbin/sshd -D

Aug 17 17:37:54 workstation systemd[1]: Starting OpenBSD Secure Shell server
Aug 17 17:37:54 workstation sshd[2653]: Server listening on 0.0.0.0 port 22.
Aug 17 17:37:54 workstation sshd[2653]: Server listening on :: port 22.
Aug 17 17:37:54 workstation systemd[1]: Started OpenBSD Secure Shell server.
```

***server1***

```
leonard@workstation:~$ sudo service ssh start
leonard@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-08-17 17:39:12 PST; 4min 34s ago
 Main PID: 2708 (sshd)
    Tasks: 1 (limit: 4884)
   CGroup: /system.slice/ssh.service
           └─2708 /usr/sbin/sshd -D

Aug 17 17:39:12 workstation sshd[2708]: Server listening on 0.0.0.0 port 22
Aug 17 17:39:12 workstation sshd[2708]: Server listening on :: port 22.
Aug 17 17:39:12 workstation systemd[1]: Started OpenBSD Secure Shell server
leonard@workstation:~$
```

***server2***

```
leonard@workstation:~$ sudo service ssh start
leonard@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-08-17 17:39:27 PST; 4min 38s ago
 Main PID: 2654 (sshd)
    Tasks: 1 (limit: 4884)
   CGroup: /system.slice/ssh.service
           └─2654 /usr/sbin/sshd -D

Aug 17 17:39:27 workstation systemd[1]: Starting OpenBSD Secure Shell serve
Aug 17 17:39:27 workstation sshd[2654]: Server listening on 0.0.0.0 port 22
Aug 17 17:39:27 workstation sshd[2654]: Server listening on :: port 22.
Aug 17 17:39:27 workstation systemd[1]: Started OpenBSD Secure Shell server
lines 1-12/12 (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*

***workstation***

```
Aug 17 17:37:54 workstation systemd[1]: Started OpenB
leonard@workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
leonard@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
leonard@workstation:~$ sudo ufw status
Status: active

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

***server1***

```
leonard@workstation:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
leonard@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
leonard@workstation:~$ sudo ufw status
Status: active

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

***server2***

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

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

leonard@workstation:~$ S
```

**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 Local Machine 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

```
leonard@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=0.932 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.522 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.503 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.641 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.452 ms
```

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

```
leonard@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=1.21 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.489 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.467 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.700 ms
```

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

```
leonard@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=4.69 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.514 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.485 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

```
leonard@workstation:~$ ssh leonard@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be establis
.
ECDSA key fingerprint is SHA256:4fxKZZzJ9yxJISLboDUcZFQEF2Yk1w3lS+mzDAKTtw8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hos
leonard@192.168.56.102's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)
```

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.

```
leonard@SERVER1:~$ logout
Connection to 192.168.56.102 closed.
leonard@workstation:~$ ssh leonard@192.168.56.103
```

```
leonard@SERVER2:~$ logout
Connection to 192.168.56.103 closed.
leonard@workstation:~$
```

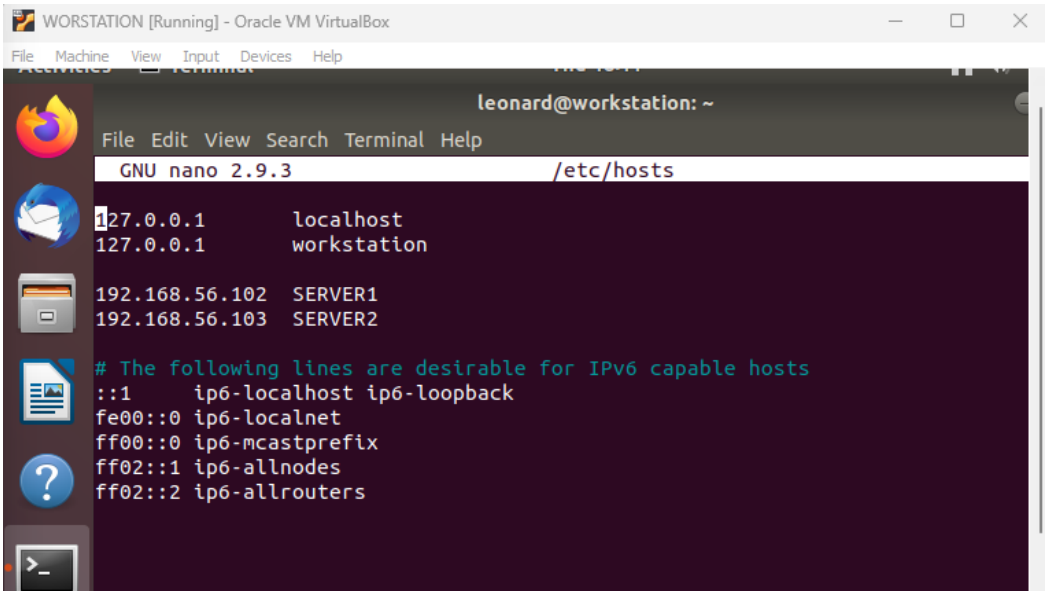
4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:

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

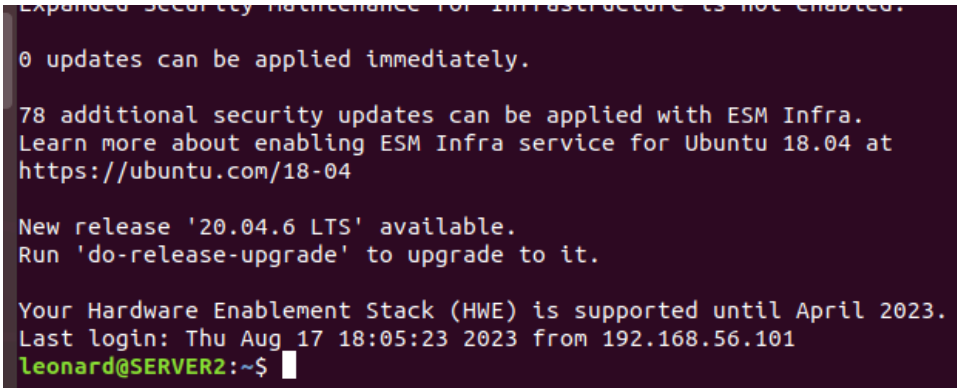
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

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



server 2

```
78 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

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 18:09:38 2023 from 192.168.56.101
leonard@SERVER1:~$
```

**Reflections:**

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   we declare ip address in shell script using SSH commands.
2. How secured is SSH?
   SSH uses encryptions to secure the connection between a client and a servers.