Name: Castillo, Joshua L.	**Date Performed: 1/16/24**
Course/Section: CPE232-CPE31S1	**Date Submitted:**
Instructor: Dr. Jonathan V. Tylar	**Semester and SY: 2023-2024**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

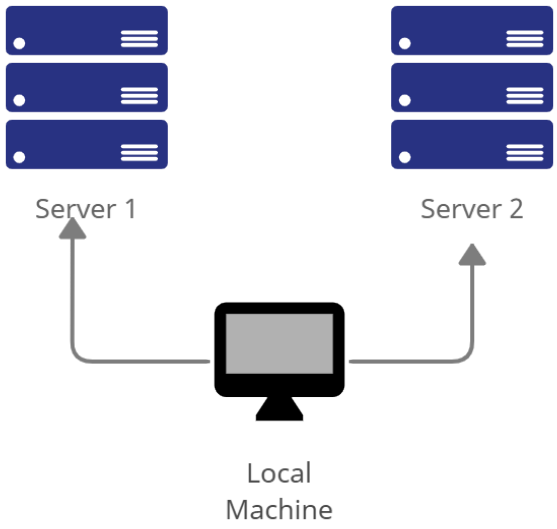
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

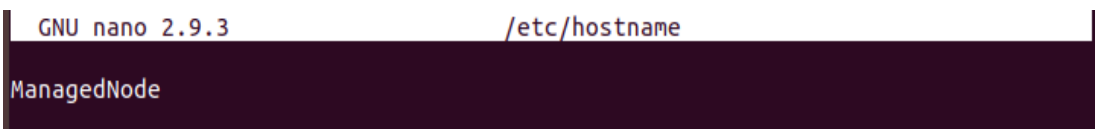
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



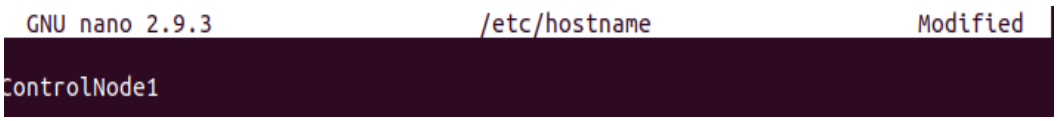
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

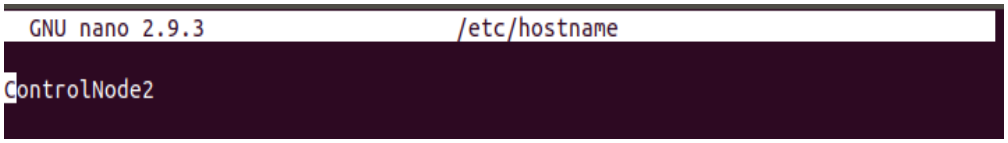
1.1 workstation for the Local Machine



1.2 Use server1 for Server 1



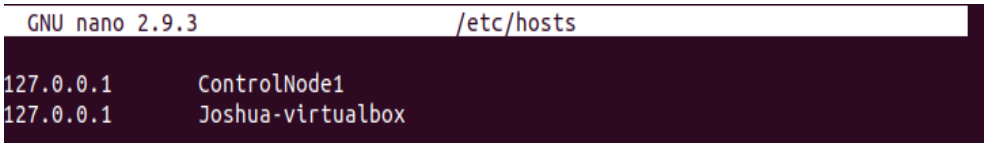
1.3 Use server2 for Server 2



Use

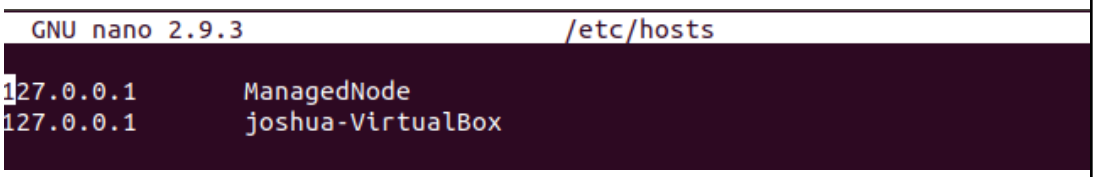
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

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 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.

```
joshua@joshua-VirtualBox:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
679 packages can be upgraded. Run 'apt list --upgradable' to see them.
joshua@joshua-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
```

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

```
joshua@joshua-VirtualBox:~$ 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
```

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*

```
joshua@joshua-VirtualBox:~$ sudo service ssh start
joshua@joshua-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
   Active: active (running) since Mon 2024-01-22 19:21:53 PST; 1min 10s ago
 Main PID: 19814 (sshd)
    Tasks: 1 (limit: 4659)
   CGroup: /system.slice/ssh.service
           └─19814 /usr/sbin/sshd -D

Jan 22 19:21:53 joshua-VirtualBox systemd[1]: Starting OpenBSD Secure Shell ser
Jan 22 19:21:53 joshua-VirtualBox sshd[19814]: Server listening on 0.0.0.0 port
Jan 22 19:21:53 joshua-VirtualBox sshd[19814]: Server listening on :: port 22.
Jan 22 19:21:53 joshua-VirtualBox systemd[1]: Started OpenBSD Secure Shell serv
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*

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

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

joshua@joshua-VirtualBox:~$
```

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: 192.168.56.123
   1.2 Server 2 IP address: 192.168.56.125
   1.3 Server 3 IP address: 192.168.56.124

2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1:

```
joshua@ManagedNode:~$ ping 192.168.56.125
PING 192.168.56.125 (192.168.56.125) 56(84) bytes of data.
64 bytes from 192.168.56.125: icmp_seq=1 ttl=64 time=0.719 ms
64 bytes from 192.168.56.125: icmp_seq=2 ttl=64 time=0.753 ms
64 bytes from 192.168.56.125: icmp_seq=3 ttl=64 time=0.454 ms
64 bytes from 192.168.56.125: icmp_seq=4 ttl=64 time=0.300 ms
64 bytes from 192.168.56.125: icmp_seq=5 ttl=64 time=0.387 ms
```

   2.2 Connectivity test for Local Machine 1 to Server 2:

```
joshua@ManagedNode:~$ ping 192.168.56.124
PING 192.168.56.124 (192.168.56.124) 56(84) bytes of data.
64 bytes from 192.168.56.124: icmp_seq=1 ttl=64 time=0.549 ms
64 bytes from 192.168.56.124: icmp_seq=2 ttl=64 time=0.430 ms
64 bytes from 192.168.56.124: icmp_seq=3 ttl=64 time=0.418 ms
^Z
```

   2.3 Connectivity test for Server 1 to Server 2:

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

```
joshua@ManagedNode:~$ ssh joshua@192.168.56.125
joshua@192.168.56.125's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-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.
*** System restart required ***

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.
```

1.3 Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*

```
joshua@ControlNode1:~$ ssh joshua@ControlNode1
The authenticity of host 'controlnode1 (127.0.0.1)' can't be established
ECDSA key fingerprint is SHA256:HIEHRFmVXfKcS6HmNHE5AMnJG0H84+cwoUNUM43r
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': y
Please type 'yes' or 'no': yes
Warning: Permanently added 'controlnode1' (ECDSA) to the list of known h
joshua@controlnode1's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-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

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.
*** System restart required ***
Last login: Mon Jan 22 20:18:44 2024 from 192.168.56.123
```

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)

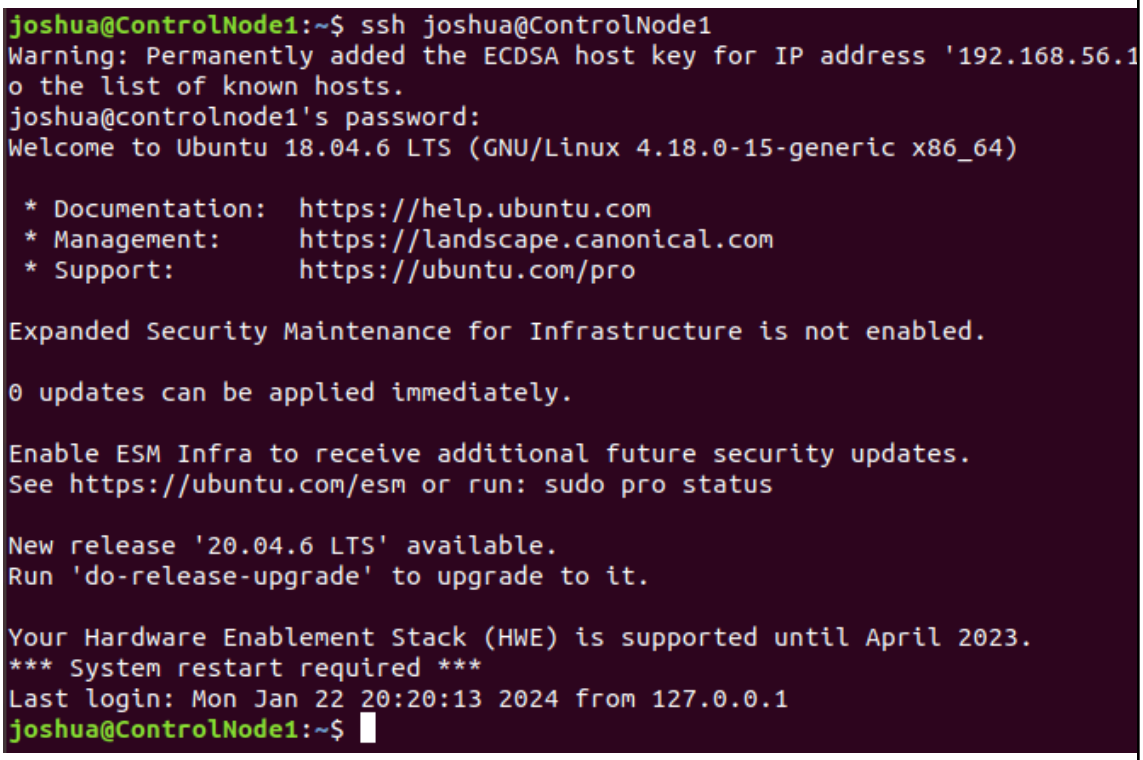
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

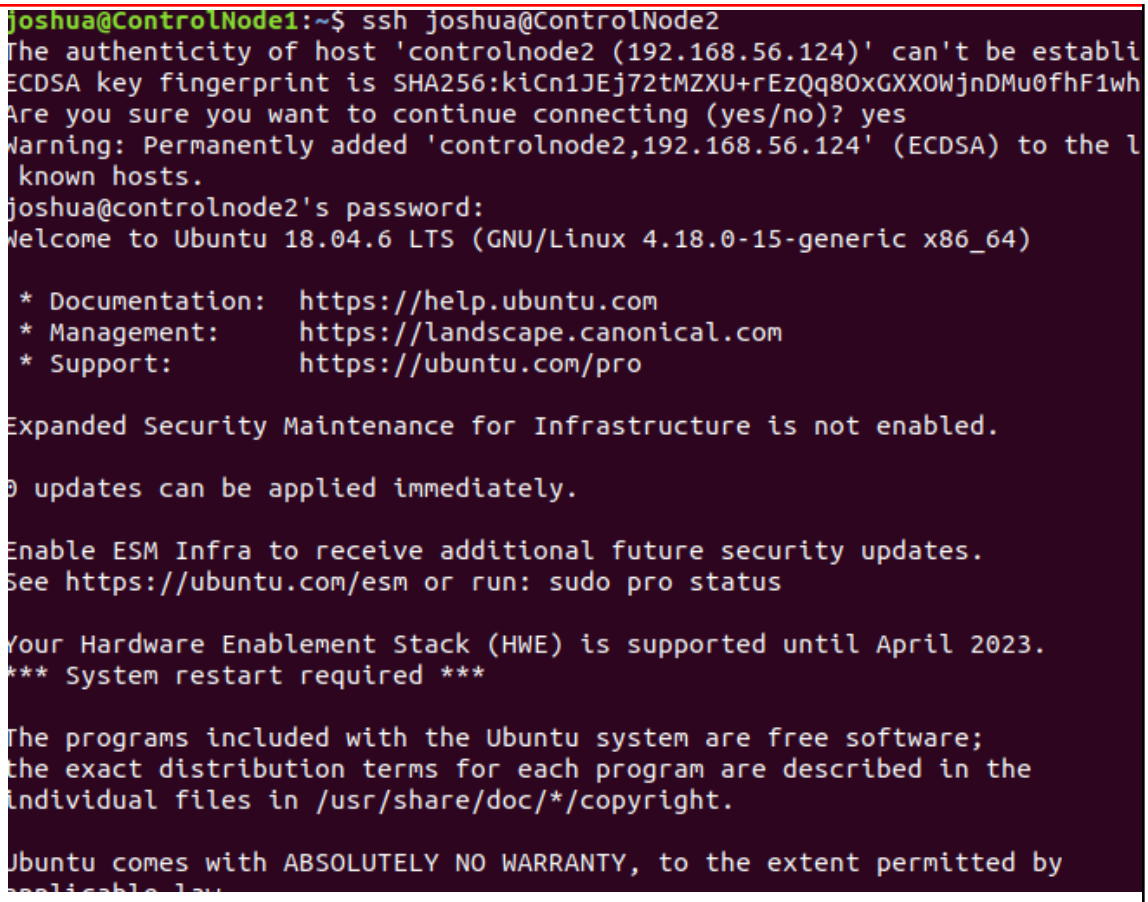
```
  GNU nano 2.9.3                        /etc/hosts

192.168.56.125  ControlNode1
192.168.56.124  ControlNode2
```

5. aptOn 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.



Reflections:

in this activity we were able to install ubuntu in virtual box and make two clones out of it with different mac addresses.

Answer the following:

1. **How are we able to use the hostname instead of IP address in SSH commands?**
   it is edited in the /etc/hosts of each server
2. **How secured is SSH?** It is secured because it is its purpose, we can have the server as it is but with SSH we can encrypt the networks used to protect its data.