Name: Ralph Andrei M. Diocera	Date Performed: 8/14/2023
Course/Section: CPE232/CPE31S4	Date Submitted: 8/15/2023
Instructor: Engr. Jonathan V. Taylar	Semester and SY: 1st Sem/2023-2024

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

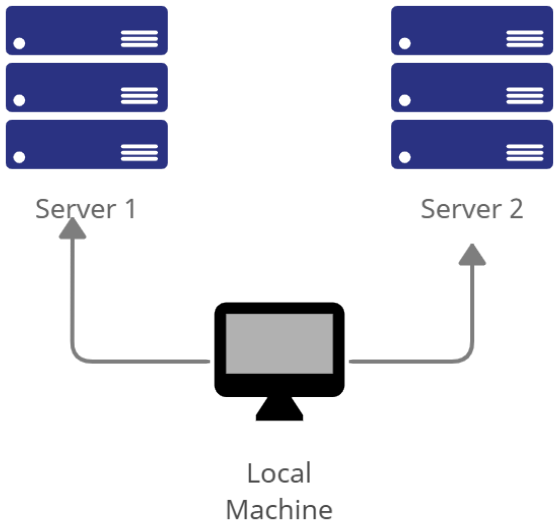
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

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

2. **Discussion:**

Network Topology:

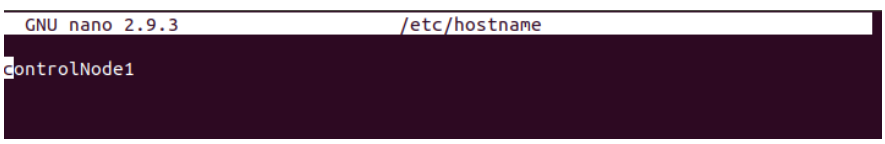
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



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

```
 GNU nano 2.9.3                    /etc/hostname

controlNode2
```

1.3 Use workstation for the Local Machine

```
 GNU nano 2.9.3                    /etc/hostname

manageNode
```

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

```
 GNU nano 2.9.3                      /etc/hosts

127.0.0.1       controlNode1
127.0.1.1       ralph-VirtualBox

# 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

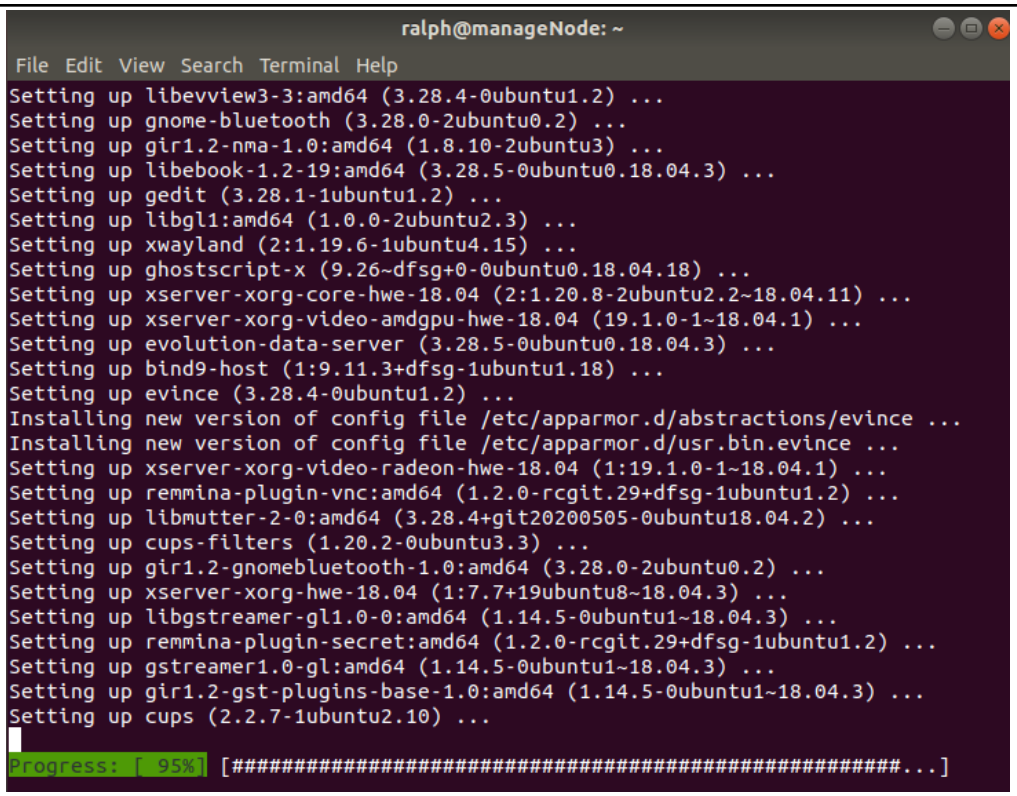
```
 GNU nano 2.9.3                      /etc/hosts

127.0.0.1       controlNode2
127.0.1.1       ralph-VirtualBox

# 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

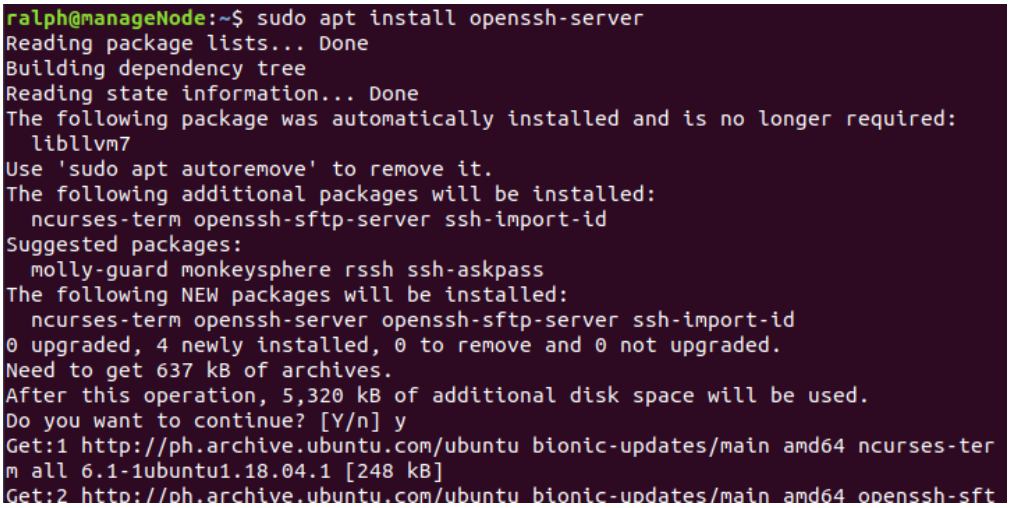
```
 GNU nano 2.9.3                      /etc/hosts

127.0.0.1       manageNode
127.0.1.1       ralph-VirtualBox

# 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
```

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*

```
ralph@manageNode:~$ sudo service ssh start
ralph@manageNode:~$ 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 Tue 2023-08-15 17:44:29 PST; 55s ago
 Main PID: 9670 (sshd)
    Tasks: 1 (limit: 2312)
   CGroup: /system.slice/ssh.service
           └─9670 /usr/sbin/sshd -D

Aug 15 17:44:29 manageNode systemd[1]: Starting OpenBSD Secure Shell server...
Aug 15 17:44:29 manageNode sshd[9670]: Server listening on 0.0.0.0 port 22.
Aug 15 17:44:29 manageNode sshd[9670]: Server listening on :: port 22.
Aug 15 17:44:29 manageNode systemd[1]: Started OpenBSD Secure Shell server.
 ESCOC
```

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*

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: 192.168.56.101
   1.2 Server 2 IP address: 192.168.56.102
   1.3 Server 3 IP address: 192.168.56.103
2. Make sure that they can ping each other.
   2.1 Connectivity test for Local Machine 1 to Server 1: ☐ Successful ☐ Not Successful

```
ralph@manageNode:~$ 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=1.15 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.640 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.85 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=1.11 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.602 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.638 ms
```

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

```
ralph@manageNode:~$ 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=0.867 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=3.17 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.05 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.35 ms
```

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

```
ralph@manageNode:~$ 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=1.29 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=1.26 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.401 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.441 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=1.24 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.432 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=0.473 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=64 time=0.892 ms
64 bytes from 192.168.56.103: icmp_seq=9 ttl=64 time=0.483 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*

```
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
ralph@192.168.56.102's password:
Permission denied, please try again.
ralph@192.168.56.102'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

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
applicable law.

ralph@manageNode:~$
```

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*

```
ralph@manageNode:~$ ssh ralph@192.168.56.102
ralph@192.168.56.102'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: Tue Aug 15 18:08:10 2023 from 192.168.56.101
ralph@controlNode1:~$
```

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

```
ralph@controlNode1:~$ logout
Connection to 192.168.56.102 closed.
ralph@manageNode:~$
```

3. Do the same for Server 2.

```
ralph@controlNode2:~$ ssh ralph@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established
.
ECDSA key fingerprint is SHA256:PVFdmtoYdktj7jG6bnHvEL6pjvKujcfxd1nxQgitFI0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
ralph@192.168.56.102's password:
Permission denied, please try again.
ralph@192.168.56.102'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: Tue Aug 15 18:11:08 2023 from 192.168.56.101
ralph@controlNode1:~$
```

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)

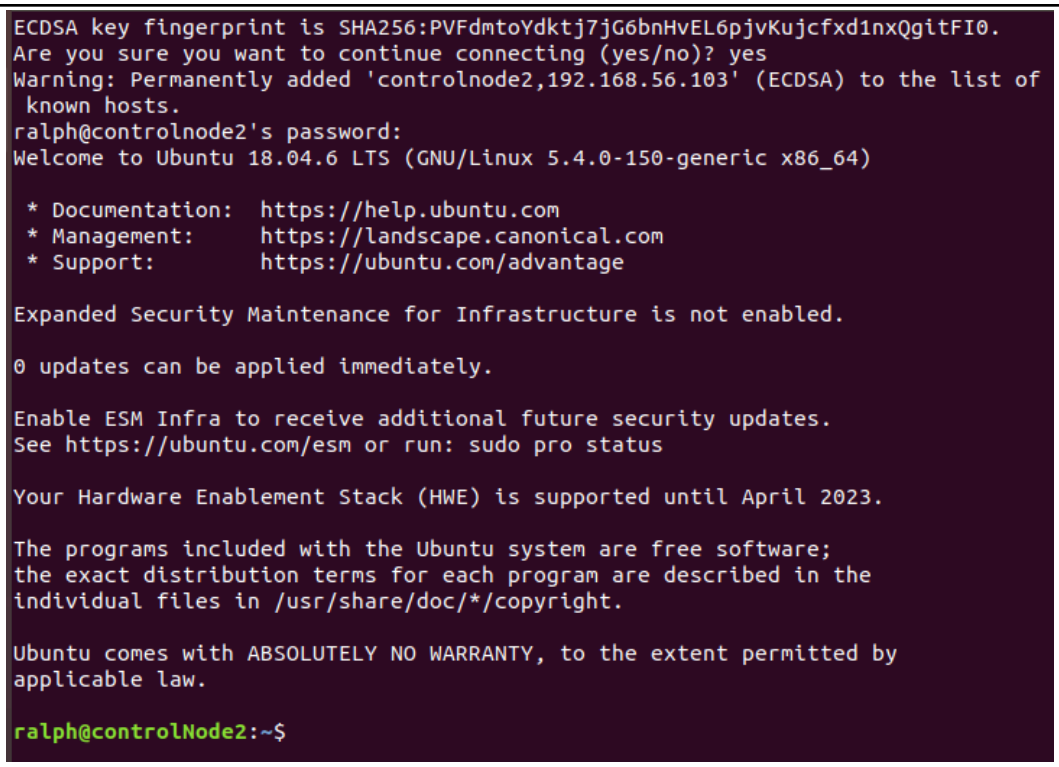
4.2 IP_address server 2 (provide the ip address of server 2 followed by the hostname)

4.3 Save the file and exit.

```
  GNU nano 2.9.3                    /etc/hosts

192.168.56.102  controlNode1
192.168.56.103  controlNode2

# 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
```

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.

```
ralph@manageNode:~$ ssh ralph@controlNode1
The authenticity of host 'controlnode1 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:PVFdmtoYdktj7jG6bnHvEL6pjvKujcfxd1nxQgitFI0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'controlnode1' (ECDSA) to the list of known hosts.
ralph@controlnode1'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: Tue Aug 15 18:12:03 2023 from 192.168.56.103
ralph@controlNode1:~$
```



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
    - In terms of SSH commands, name resolution is an efficient way of being used with the hostname instead of the IP address provided because of the given process that maps the name over the IP address provided. Before going to the process of using the hostname, it needs a name resolution or a format like “ssh hostname.local” whereas the local option because it is being run into the local network.
2. How secured is SSH?
    - In terms of the SSH’s security, it has established or incorporates encryption that lets the user ensure that no hacking will be implemented that can interpret the traffic between two connected devices because within the SSH channel, it has been given that in order to access within, it needs a public-key cryptography that lets the client authenticate which makes the actions like managing and other priorities will remain private.

Conclusion:

- In conclusion, Virtual Box has been a flexible platform for creating and configuring different types of Virtual Machines whereas a single channel made as the main workstation can be cloned in any quantity with current progresses that can be labeled as the server that generates a different MAC Address from the main channel. I have also learned that SSH commands can be used using channels that revolve within the Virtual Box as one group like a secure remote access that changes the host in an instance like how we use the “ssh user@host” command that establishes a secure remote shell session over the remote host. The SSH or Secure Shell is truly an essential feature of Linux for how it is considered as a key tool and as a user-managing feature for administrators and developers.