

Name: Rio Marie G. Suzuki	Date Performed: 08/17/2023
Course/Section: CPE232S8	Date Submitted: 08/17/2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem 2023-2024

Activity 1: Configure Network using Virtual Machines

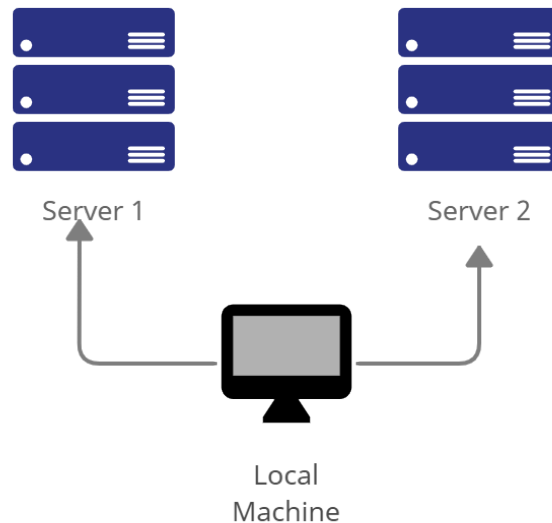
1. Objectives:

- 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*).



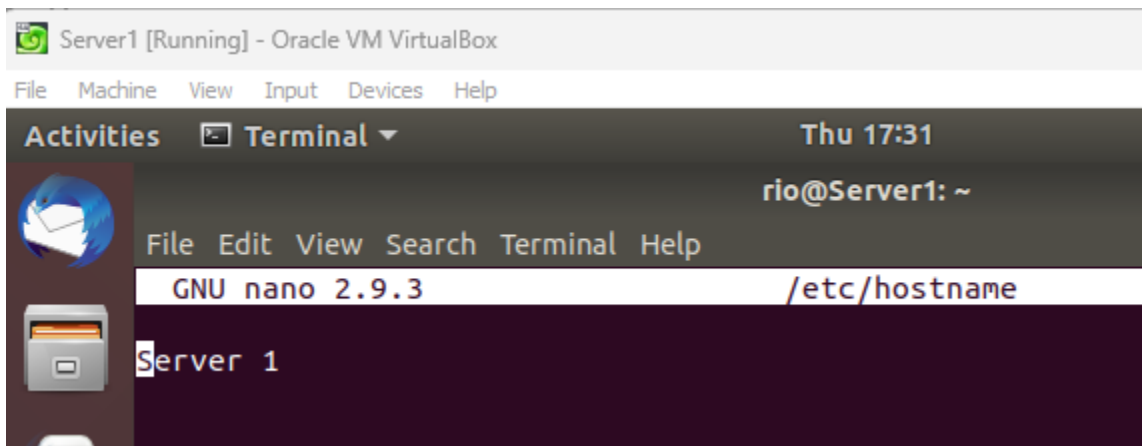
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

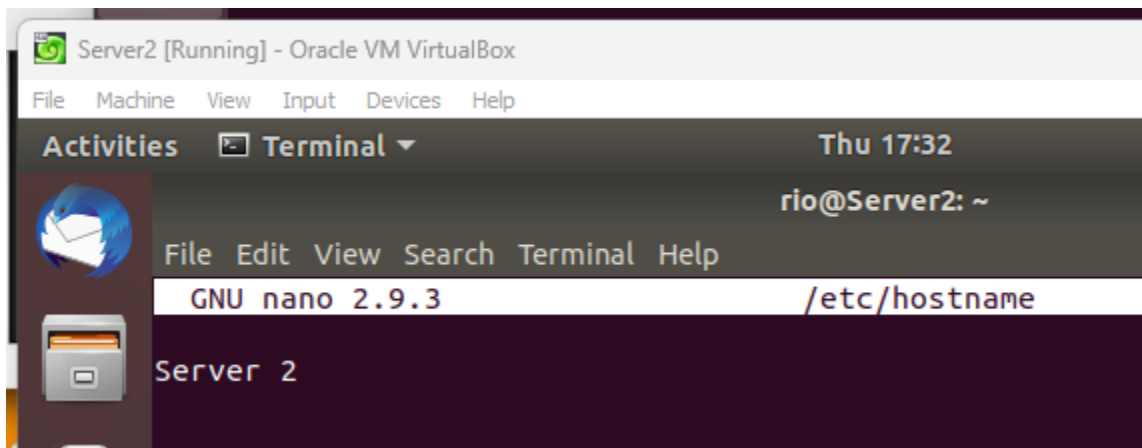
```

File Edit View Search Terminal Help
rio@Workstation:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Workstation:~$
  
```

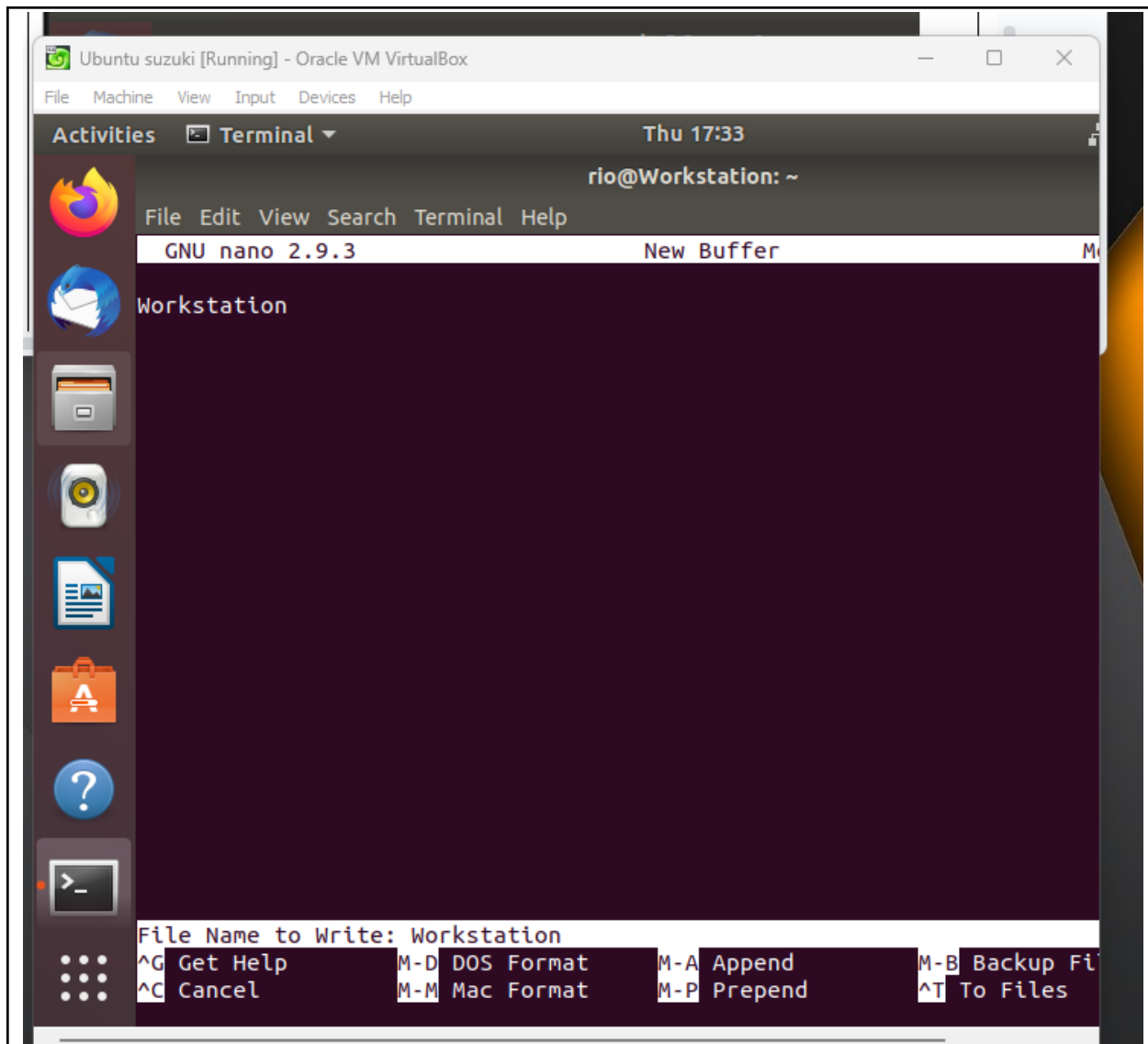
1.1 Use server1 for Server 1



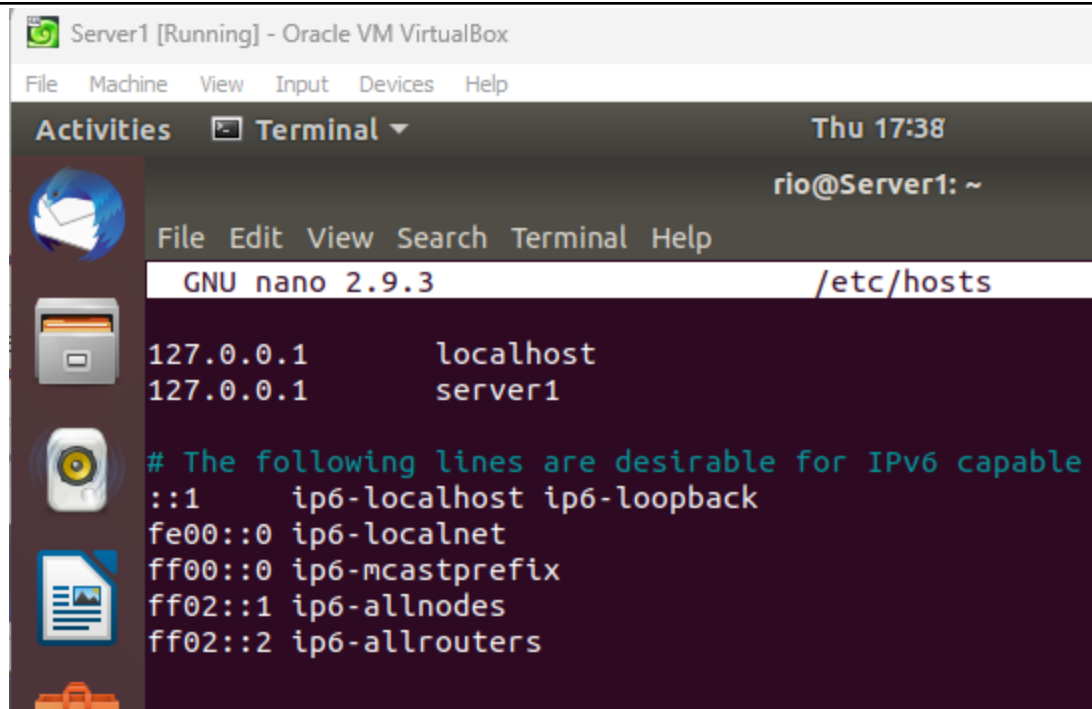
1.2 Use server2 for Server 2



1.3 Use workstation for the Local Machine

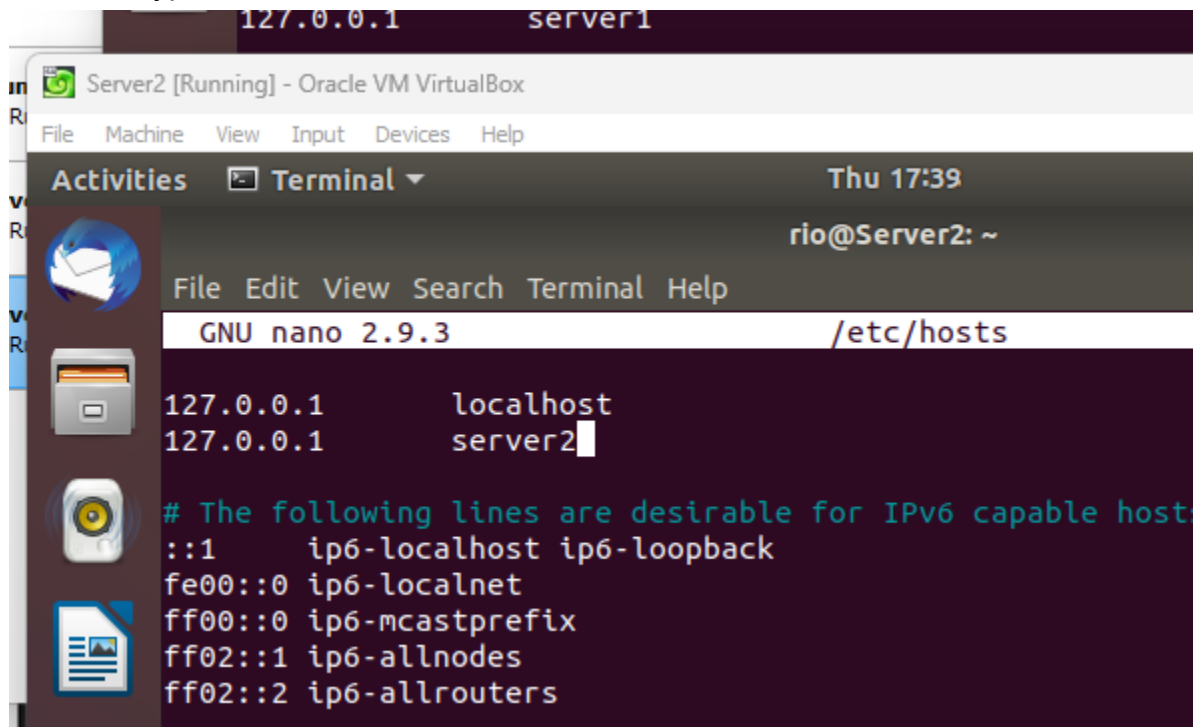


2. Edit the hosts using the command `sudo nano /etc/hosts`. Edit the second line.
2.1 Type 127.0.0.1 server 1 for Server 1



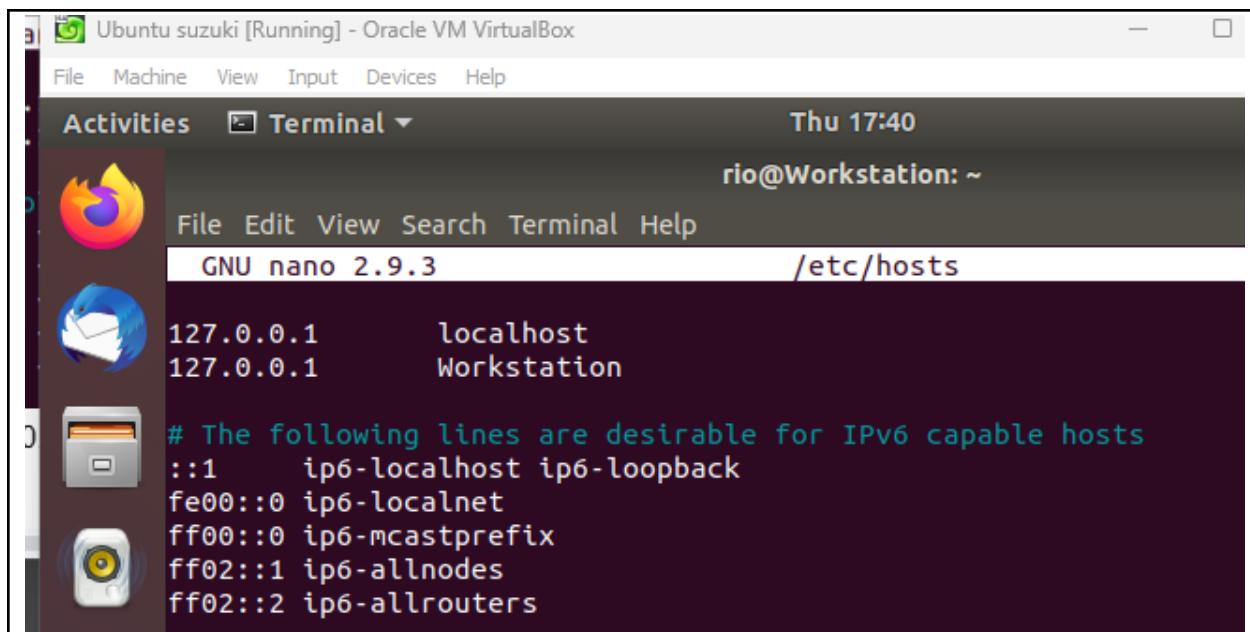
```
Server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:38
rio@Server1: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
127.0.0.1 server1
# The following lines are desirable for IPv6 capable
::1 ip6-localhost ip6-loopback
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



```
Server2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:39
rio@Server2: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
127.0.0.1 server2
# The following lines are desirable for IPv6 capable host
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

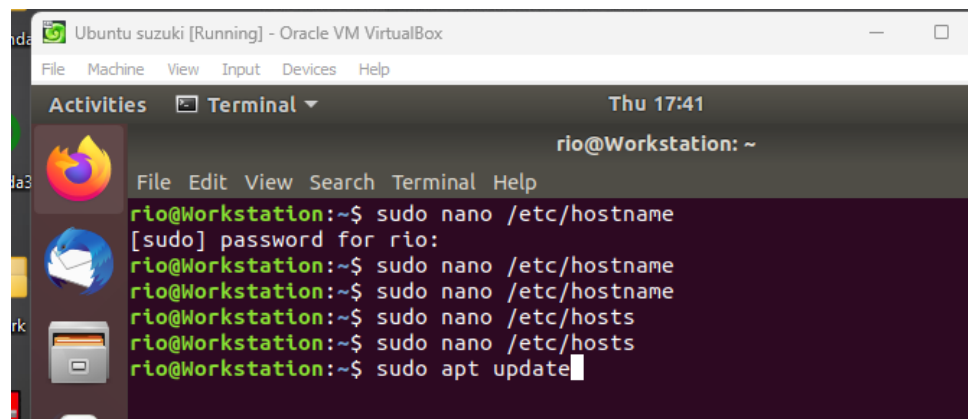
2.3 Type 127.0.0.1 workstation for the Local Machine



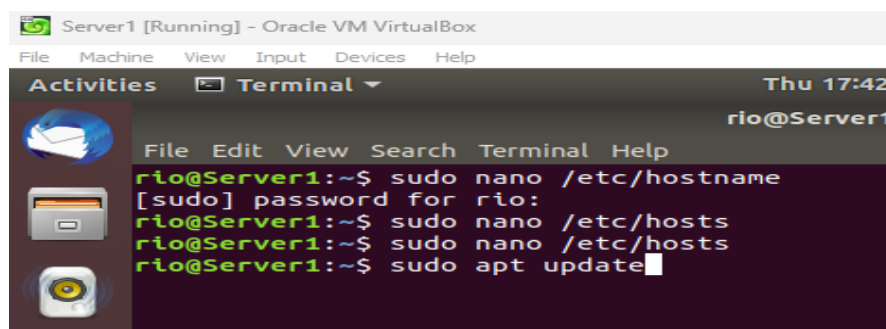
```
Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:40
rio@Workstation: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
127.0.0.1 Workstation
# The following lines are desirable for IPv6 capable hosts
::1 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.



```
Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:41
rio@Workstation: ~
File Edit View Search Terminal Help
rio@Workstation:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Workstation:~$ sudo nano /etc/hostname
rio@Workstation:~$ sudo nano /etc/hostname
rio@Workstation:~$ sudo nano /etc/hosts
rio@Workstation:~$ sudo nano /etc/hosts
rio@Workstation:~$ sudo apt update
```



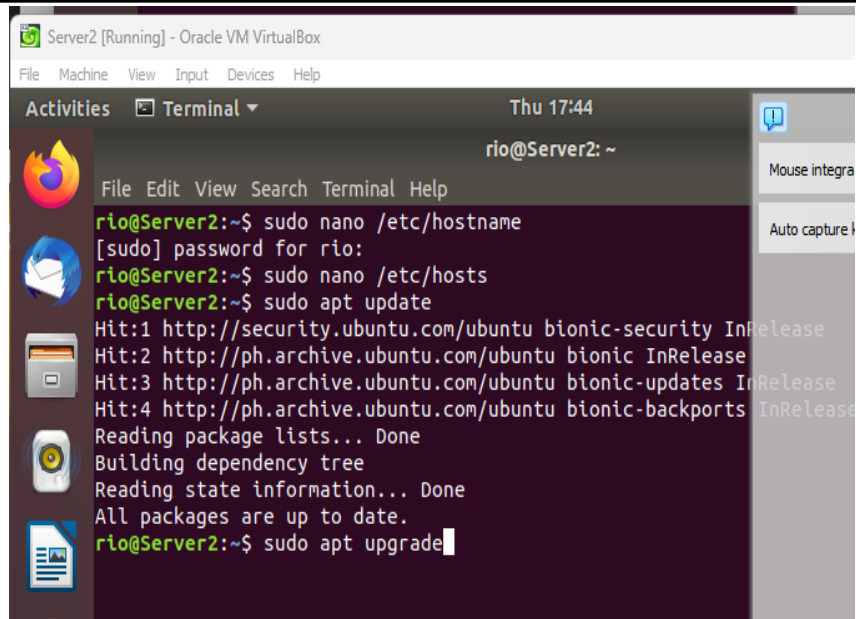
```
Server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:42
rio@Server1: ~
File Edit View Search Terminal Help
rio@Server1:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Server1:~$ sudo nano /etc/hosts
rio@Server1:~$ sudo nano /etc/hosts
rio@Server1:~$ sudo apt update
```

```
Server2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:42
rio@Server2: ~
File Edit View Search Terminal Help
rio@Server2:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Server2:~$ sudo nano /etc/hosts
rio@Server2:~$ sudo apt update
```

----- Upgrade -----

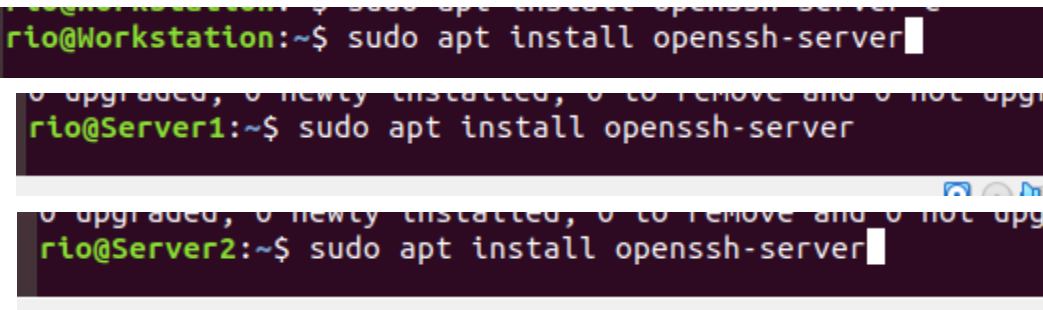
```
Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:43
rio@Workstation: ~
File Edit View Search Terminal Help
rio@Workstation:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Workstation:~$ sudo nano /etc/hostname
rio@Workstation:~$ sudo nano /etc/hostname
rio@Workstation:~$ sudo nano /etc/hosts
rio@Workstation:~$ sudo nano /etc/hosts
rio@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.
rio@Workstation:~$ sudo apt upgrade
```

```
Server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:43
rio@Server1: ~
File Edit View Search Terminal Help
rio@Server1:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Server1:~$ sudo nano /etc/hosts
rio@Server1:~$ sudo nano /etc/hosts
rio@Server1:~$ 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.
rio@Server1:~$ sudo apt upgrade
```



```
Server2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 17:44
rio@Server2: ~
File Edit View Search Terminal Help
rio@Server2:~$ sudo nano /etc/hostname
[sudo] password for rio:
rio@Server2:~$ sudo nano /etc/hosts
rio@Server2:~$ 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.
rio@Server2:~$ sudo apt upgrade
```

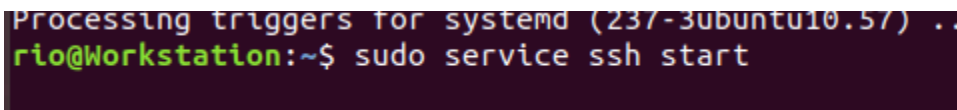
2. Install the SSH server using the command *sudo apt install openssh-server*.



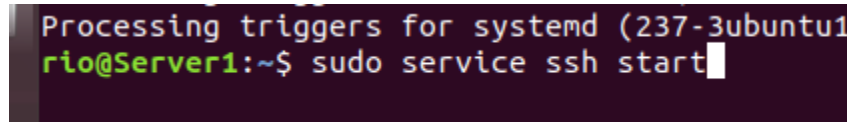
```
rio@Workstation:~$ sudo apt install openssh-server
rio@Server1:~$ sudo apt install openssh-server
rio@Server2:~$ sudo apt install openssh-server
```

3. Verify if the SSH service has started by issuing the following commands:

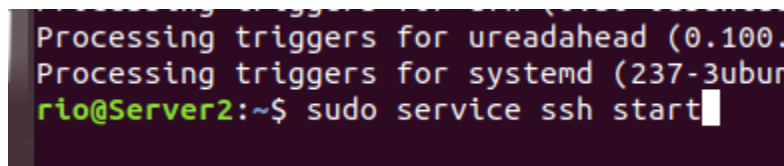
- 3.1 *sudo service ssh start*



```
Processing triggers for systemd (237-3ubuntu10.57) ..
rio@Workstation:~$ sudo service ssh start
```



```
Processing triggers for systemd (237-3ubuntu10.57) ..
rio@Server1:~$ sudo service ssh start
```



```
Processing triggers for ureadahead (0.100.0-1ubuntu0.1) ..
Processing triggers for systemd (237-3ubuntu10.57) ..
rio@Server2:~$ sudo service ssh start
```

3.2 *sudo systemctl status ssh*

```
rio@Workstation:~$ sudo service ssh start
rio@Workstation:~$ sudo systemctl status ssh
```

```
Processing triggers for systemd (237-3ubuntu10.57) ...
```

```
rio@Server1:~$ sudo service ssh start
rio@Server1:~$ sudo systemctl status ssh
```

```
rio@Server2:~$ sudo service ssh start
rio@Server2:~$ sudo systemctl status ssh
```

-----Output-----

```
rio@Workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-08-17 17:47:41 PST; 3min 7s
     Main PID: 2686 (sshd)
        Tasks: 1 (limit: 4656)
      CGroup: /system.slice/ssh.service
              └─2686 /usr/sbin/sshd -D

Aug 17 17:47:41 Workstation systemd[1]: Starting OpenBSD Secure Shell server: sshd.
Aug 17 17:47:41 Workstation sshd[2686]: Server listening on 0.0.0.0 port 22.
Aug 17 17:47:41 Workstation sshd[2686]: Server listening on :: port 22.
Aug 17 17:47:41 Workstation systemd[1]: Started OpenBSD Secure Shell server: sshd.
rio@Workstation:~$
```

4. Configure the firewall to all port 22 by issuing the following commands:

4.1 *sudo ufw allow ssh*

```
rio@Workstation:~$ sudo ufw allow ssh
```

```
rio@Workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
rio@Workstation:~$
```

```
Aug 17 17:47:44 Server1 sshd[2621]: Server listening on 0.0.0.0 port 22.
rio@Server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
rio@Server1:~$
```



```
rio@Server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
rio@Server2:~$
```

4.2 *sudo ufw enable*

```
rio@Workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
rio@Workstation:~$
```

4.3 *sudo ufw status*

```
Firewall is active and enabled on system startup
rio@Workstation:~$ sudo ufw status
Status: active

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

rio@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: 192.168.56.102

1.2 Server 2 IP address: 192.168.56.103

1.3 Server 3 IP address: 192.168.56.101

```
rio@Workstation:~$ ifconfig

Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools

rio@Workstation:~$ sudo apt install net-tools
```

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::b168:1616:9b46:f23e prefixlen 64 scopeid 0x20<link>
ether 08:00:27:fd:f3:63 txqueuelen 1000 (Ethernet)
RX packets 319 bytes 43043 (43.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 84 bytes 10782 (10.7 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::dbc9:6874:f6b2:15e8 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:3a:94:97 txqueuelen 1000 (Ethernet)
RX packets 118 bytes 15998 (15.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 73 bytes 9270 (9.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255
inet6 fe80::c2d9:7253:3ab:8df0 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:19:86:1b txqueuelen 1000 (Ethernet)
RX packets 113 bytes 15508 (15.5 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 78 bytes 9928 (9.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
```

2. Make sure that they can ping each other.

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

```
[1]+ Stopped ping 192.168.56.103
rio@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=0.763 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.815 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.812 ms
```

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

```
rio@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=1.28 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.392 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.567 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=1.42 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.921 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=1.27 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=64 time=0.923 ms
64 bytes from 192.168.56.103: icmp_seq=9 ttl=64 time=0.876 ms
64 bytes from 192.168.56.103: icmp_seq=10 ttl=64 time=0.521 ms
64 bytes from 192.168.56.103: icmp_seq=11 ttl=64 time=1.65 ms
```

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

```
rio@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=0.932 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=1.56 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.420 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.860 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=1.02 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=1.09 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=64 time=0.576 ms
64 bytes from 192.168.56.103: icmp_seq=9 ttl=64 time=1.15 ms
64 bytes from 192.168.56.103: icmp_seq=10 ttl=64 time=1.29 ms
64 bytes from 192.168.56.103: icmp_seq=11 ttl=64 time=1.17 ms
64 bytes from 192.168.56.103: icmp_seq=12 ttl=64 time=1.40 ms
64 bytes from 192.168.56.103: icmp_seq=13 ttl=64 time=1.63 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 jvtaylor@192.168.56.120`

```
[2]+  Stopped                  ping 192.168.56.102
rio@Workstation:~$ ssh rio@192.168.56.102
```

```

[1] - stopped ping server1
rio@Workstation:~$ ssh rio@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:NknGf1NsDB2VW0Y1ejfEz/JO6HB5RUK9H3uNey8renk.
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.
rio@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.

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

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.

```

1.2 Enter the password for server 1 when prompted

```

Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
rio@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

```

1.3 Verify that you are in server 1. The user should be in this format user@server1.

For example, *jvtaylor@server1*

```

rio@Server1:~$ ssh rio@server1
The authenticity of host 'server1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:NknGf1NsDB2VW0Y1ejfEz/JO6HB5RUK9H3uNey8renk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1' (ECDSA) to the list of known hosts.
rio@server1's password:

```

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

```

Last login: Thu Aug 17 18:09:23 2023
rio@Server1:~$ logout
Connection to server1 closed.
rio@Server1:~$

```

3. Do the same for Server 2.

```

Last login: Thu Aug 17 18:16:00 2023
rio@Server2:~$ logout
Connection to server2 closed.
rio@Server2:~$

```

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

```
Connection to server1 closed.  
rio@Server1:~$ sudo nano /etc/hosts  
[sudo] password for rio:
```

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

```
1 127.0.0.1      localhost  
127.0.0.1      server1  
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
```

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

```
1 127.0.0.1      localhost  
127.0.0.1      server1  
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 jvtaylor@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

```
rio@Server1:~$ ssh rio@server1
```

```
rio@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.

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 18:14:07 2023 from 127.0.0.1
rio@Server1:~$
```

```
Last login: Thu Aug 17 18:14:07 2023 from 127.0.0.1
rio@Server1:~$ ssh rio@server2
The authenticity of host 'server2 (192.168.56.103)' can't be established.
ECDSA key fingerprint is SHA256:cLt/EfW6A/Wnb55bR00Y8mvrD09LtzUMiAerQGy41YU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server2,192.168.56.103' (ECDSA) to the list of know
n hosts.
rio@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.

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 18:23:45 2023 from 127.0.0.1
rio@Server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
 - As I do the activity I can say that placing a hostname instead of using an IP address in SSH command is way easier since sometimes we forget the ip since it is a combination of just numbers which for some people like me is hard to remember. Also, sometimes in some cases we typed the wrong number in IP and it is not interchangeable so it would be a hassle work. What we did to use a hostname instead of an IP is the DNS resolution where it converts the IP into hostname for easier access.
2. How secured is SSH?
 - SSH or Secure shell is one of the most considered secured shell in the world because it uses encryption in passwords and public keys for authentication. One of the best feature of ssh is the mechanism where it can detect if the data transferred is changed or not and if it is really altered during the process the system will deny it.

Conclusion

To conclude this activity aims for the student to have or replenish the knowledge we have in using the Virtualbox at the same time to learn and review the basic commands in SSH. It also test the student to have the basic knowledge in solving different problems that they'll encounter during the process like getting the ip addresses etc,. This activity helps the student to learn how to set up the virtualbox and the needed contents of the terminal. I also learned how to test the connectivity of the virtual machines using the servers and the provided IP addresses.