

Name: Renier L. Lope	Date Performed: 08/25/2024
Course/Section: CPE212 – CPE31S2	Date Submitted: 08/25/2024
Instructor: Robin Valenzuela	Semester and SY: 1st Sem (2024-2025)

Activity 1: Configure Network using Virtual Machines

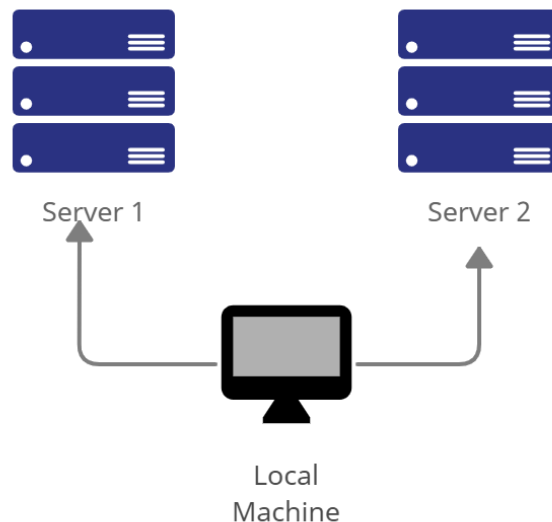
1. Objectives:

- 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

```

rnrlope@rener:~$ sudo nano /etc/hostname
[sudo] password for rnrlope:
rnrlope@rener:~$ cat /etc/hostname
server1
rnrlope@rener:~$
  
```

1.2 Use server2 for Server 2

```
rnrlope@rener:~$ sudo nano /etc/hostname
[sudo] password for rnrlope:
rnrlope@rener:~$ cat /etc/hostname
server2
rnrlope@rener:~$
```

1.3 Use workstation for the Local Machine

```
rnrlope@rener:~$ sudo nano /etc/hostname
rnrlope@rener:~$ cat /etc/hostname
workstation
rnrlope@rener:~$
```

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

```
rnrlope@rener:~$ sudo nano /etc/hosts
[sudo] password for rnrlope:
rnrlope@rener:~$ cat /etc/hosts
127.0.0.1 server1
127.0.1.1 rener

# 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
rnrlope@rener:~$
```

2.2 Type 127.0.0.1 server 2 for Server 2

```
rnrlope@rener:~$ sudo nano /etc/hosts
[sudo] password for rnrlope:
rnrlope@rener:~$ cat /etc/hosts
127.0.0.1 server2
127.0.1.1 rener

# 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
rnrlope@rener:~$
```

2.3 Type 127.0.0.1 workstation for the Local Machine

```
rnrlope@rener:~$ sudo nano /etc/hosts
rnrlope@rener:~$ cat /etc/hosts
127.0.0.1 workstation
127.0.1.1 rener

# 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
rnrlope@rener:~$
```

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.

```
rnrlope@server1:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
14 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  e2fsprogs e2fsprogs-l10n gnome-text-editor language-pack-en language-pack-en-base
  language-pack-gnome-en-base libcom-err2 libdeflate0 libext2fs2t64 libss2 logsave
  python-apt-common python3-apt
The following packages have been kept back:
  language-pack-gnome-en
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
N: Some packages may have been kept back due to phasing.
rnrlope@server1:~$
```

Figure 2.1.1 `sudo apt update && sudo apt upgrade (server1)`

```

rnrlope@server2:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
14 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  e2fsprogs e2fsprogs-l10n gnome-text-editor language-pack-en language-pack-en-base
  language-pack-gnome-en-base libcom-err2 libdeflate0 libext2fs2t64 libss2 logsave
  python-apt-common python3-apt
The following packages have been kept back:
  language-pack-gnome-en
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
N: Some packages may have been kept back due to phasing.
rnrlope@server2:~$

```

Figure 2.1.2 `sudo apt update && sudo apt upgrade (server2)`

```

rnrlope@workstation:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ph.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
14 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  e2fsprogs e2fsprogs-l10n gnome-text-editor language-pack-en language-pack-en-base
  language-pack-gnome-en-base libcom-err2 libdeflate0 libext2fs2t64 libss2 logsave
  python-apt-common python3-apt
The following packages have been kept back:
  language-pack-gnome-en
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
N: Some packages may have been kept back due to phasing.
rnrlope@workstation:~$

```

Figure 2.1.3 `sudo apt update && sudo apt upgrade (workstation)`

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

```
rnrlope@server1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
rnrlope@server1:~$
```

Figure 2.2.1 sudo apt install openssh-server (server1)

```
rnrlope@server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
rnrlope@server2:~$
```

Figure 2.2.2 sudo apt install openssh-server (server2)

```
rnrlope@workstation:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
rnrlope@workstation:~$
```

Figure 2.2.3 sudo apt install openssh-server (workstation)

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*

```

rnrlope@server1:~$ sudo service ssh start
rnrlope@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Sun 2024-08-25 21:57:01 PST; 11s ago
 TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Process: 3845 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3847 (sshd)
      Tasks: 1 (limit: 4615)
     Memory: 1.2M (peak: 1.5M)
        CPU: 41ms
    CGroup: /system.slice/ssh.service
            └─3847 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 21:57:01 server1 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server>
Aug 25 21:57:01 server1 sshd[3847]: Server listening on :: port 22.
Aug 25 21:57:01 server1 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
rnrlope@server1:~$

```

Figure 2.3.1 sudo service ssh start & sudo systemctl status ssh (server1)

```

rnrlope@server2:~$ sudo service ssh start
rnrlope@server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Sun 2024-08-25 21:57:33 PST; 10min ago
 TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Process: 3809 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3811 (sshd)
      Tasks: 1 (limit: 4615)
     Memory: 1.2M (peak: 1.6M)
        CPU: 48ms
    CGroup: /system.slice/ssh.service
            └─3811 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 21:57:32 server2 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server>
Aug 25 21:57:33 server2 sshd[3811]: Server listening on :: port 22.
Aug 25 21:57:33 server2 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
rnrlope@server2:~$

```

Figure 2.3.2 sudo service ssh start & sudo systemctl status ssh (server2)


```

rnrlope@workstation:~$ sudo service ssh start
rnrlope@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Sun 2024-08-25 21:55:52 PST; 11min ago
 TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Process: 3700 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3702 (sshd)
      Tasks: 1 (limit: 4615)
     Memory: 1.2M (peak: 1.5M)
        CPU: 62ms
    CGroup: /system.slice/ssh.service
            └─3702 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 21:55:52 workstation systemd[1]: Starting ssh.service - OpenBSD Secure Shell se>
Aug 25 21:55:52 workstation sshd[3702]: Server listening on :: port 22.
Aug 25 21:55:52 workstation systemd[1]: Started ssh.service - OpenBSD Secure Shell ser>
rnrlope@workstation:~$

```

Figure 2.3.3 sudo service ssh start & sudo systemctl status ssh (workstation)

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*

```

rnrlope@server1:~$ sudo ufw allow ssh
[sudo] password for rnrlope:
Rules updated
Rules updated (v6)
rnrlope@server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
rnrlope@server1:~$ sudo ufw status
Status: active

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

rnrlope@server1:~$

```

Figure 2.4.1 sudo ufw allow ssh & sudo ufw enable & sudo ufw status (server1)

```

rnrlope@server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
rnrlope@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
rnrlope@server2:~$ sudo ufw status
Status: active

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

rnrlope@server2:~$ █

```

Figure 2.4.2 sudo ufw allow ssh & sudo ufw enable & sudo ufw status (server2)

```

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

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

rnrlope@workstation:~$ █

```

Figure 2.4.3 sudo ufw allow ssh & sudo ufw enable & sudo ufw status (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: 192.168.56.104

```
rnrlope@server1:~$ ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::deab:4c46:379e:984f prefixlen 64 scopeid 0x20<link>
```

1.2 Server 2 IP address: 192.168.56.105

```
rnrlope@server2:~$ ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.105 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::e5b9:6e2:be5d:9c99 prefixlen 64 scopeid 0x20<link>
```

1.3 Local Host IP address: 192.168.56.103

```
rnrlope@workstation:~$ ifconfig
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::ac8c:898c:f186:7a30 prefixlen 64 scopeid 0x20<link>
```

2. Make sure that they can ping each other.

2.1 Connectivity test for Local Machine 1 to Server 1:

☒ Successful ☐ Not Successful

```
rnrlope@workstation:~$ ping 192.168.56.104 -c 4
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=0.398 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.450 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.395 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.439 ms

--- 192.168.56.104 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.395/0.420/0.450/0.024 ms
rnrlope@workstation:~$
```

2.2 Connectivity test for Local Machine 1 to Server 2:

☒ Successful ☐ Not Successful

```
rnrlope@workstation:~$ ping 192.168.56.105 -c 4
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=0.781 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.451 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.370 ms
64 bytes from 192.168.56.105: icmp_seq=4 ttl=64 time=0.389 ms

--- 192.168.56.105 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3165ms
rtt min/avg/max/mdev = 0.370/0.497/0.781/0.166 ms
rnrlope@workstation:~$
```

2.3 Connectivity test for Server 1 to Server 2:

☒ Successful ☐ Not Successful

```
rnrlope@server1:~$ ping 192.168.56.105 -c 4
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=0.469 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.590 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.430 ms
64 bytes from 192.168.56.105: icmp_seq=4 ttl=64 time=0.445 ms

--- 192.168.56.105 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3094ms
rtt min/avg/max/mdev = 0.430/0.483/0.590/0.063 ms
rnrlope@server1:~$
```

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`

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, `jvtaylor@server1`

```
rnrlope@workstation:~$ ssh rnrlope@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:ruZzw/ugrcGwoMUv5tk/KKA1uE0dILH9oKaXzsqzhac.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known hosts.
rnrlope@192.168.56.104's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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.

rnrlope@server1:~$
```

Figure 4.1.1 ssh command for server1

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

```
rnrlope@server1:~$
logout
Connection to 192.168.56.104 closed.
rnrlope@workstation:~$
```

Figure 4.2.1 logout of server1

3. Do the same for Server 2.

```
rnrlope@workstation:~$ ssh rnrlope@192.168.56.105
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established.
ED25519 key fingerprint is SHA256:ruZzw/ugrcGwoMUv5tk/KKA1uE0dILH9oKaXzsqzhac.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.105' (ED25519) to the list of known hosts.
rnrlope@192.168.56.105's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

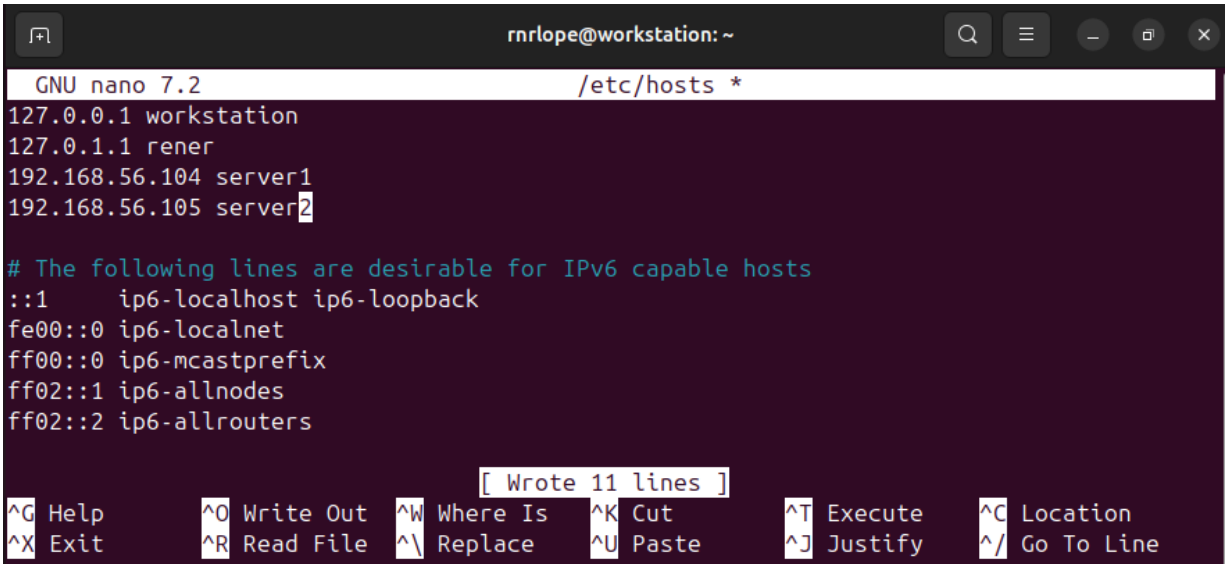
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

rnrlope@server2:~$
logout
Connection to 192.168.56.105 closed.
rnrlope@workstation:~$
```

Figure 4.3.1 ssh command for server2

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)



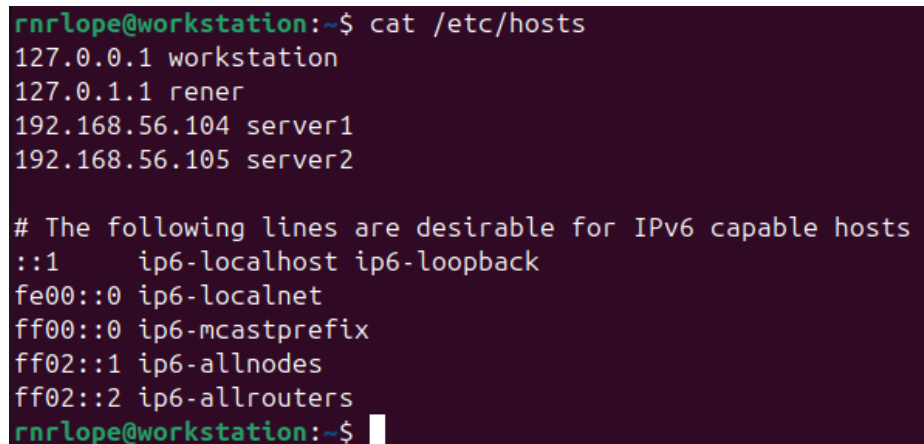
```
GNU nano 7.2 /etc/hosts *
127.0.0.1 workstation
127.0.1.1 rener
192.168.56.104 server1
192.168.56.105 server2

# 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

[ Wrote 11 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Figure 4.4.1 edit host

- 4.3 Save the file and exit.



```
rnrlope@workstation:~$ cat /etc/hosts
127.0.0.1 workstation
127.0.1.1 rener
192.168.56.104 server1
192.168.56.105 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
rnrlope@workstation:~$
```

Figure 4.4.2 verify the hosts

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.

```
rnrlope@workstation:~$ ssh rnrlope@server1
The authenticity of host 'server1 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:ruZzw/ugrcGwoMUv5tk/KKA1uE0dILH9oKaXzsqzhac.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server1' (ED25519) to the list of known hosts.
rnrlope@server1's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
connection or proxy settings

Last login: Sun Aug 25 23:19:24 2024 from 192.168.56.103
rnrlope@server1:~$
logout
Connection to server1 closed.
```

Figure 4.5.1 verify ssh command for server1

```
rnrllope@workstation:~$ ssh rnrllope@server2
The authenticity of host 'server2 (192.168.56.105)' can't be established.
ED25519 key fingerprint is SHA256:ruZzw/ugrcGwoMUv5tk/KKA1uE0dILH9oKaXzsqqzac.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
  ~/.ssh/known_hosts:6: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2' (ED25519) to the list of known hosts.
rnrllope@server2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
connection or proxy settings

Last login: Sun Aug 25 23:13:21 2024 from 192.168.56.103
rnrllope@server2:~$
logout
Connection to server2 closed.
```

Figure 4.5.2 verify ssh command for server2

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
 - Editing /etc/hosts and inputting the IP address followed by the hostname will make using SSH commands easier, instead of using the IP Address we can use the assigned hostname of the IP address.
2. How secured is SSH?
 - Secure Shell, also known as SSH, is a kind of network communication protocol that enables communication between two or more computers, primarily between SSH Client and Server. It encrypts data to prevent unwanted access from unidentified sources and uses the SHA-2 hash algorithm to render it unreadable and un-hackable.