Name: Caro, Aaron Martin P.	**Date Performed: 22/08/2023**
Course/Section: CPE232 CPE22S3	**Date Submitted: 22/08/2023**
Instructor: Sir, Roman Richard	**Semester and SY: 1st sem 2023-2024**

Activity 1: Configure Network using Virtual Machines

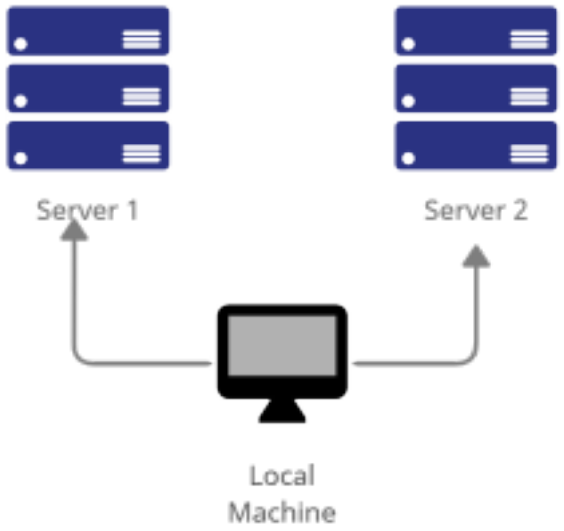
1. Objectives:

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:

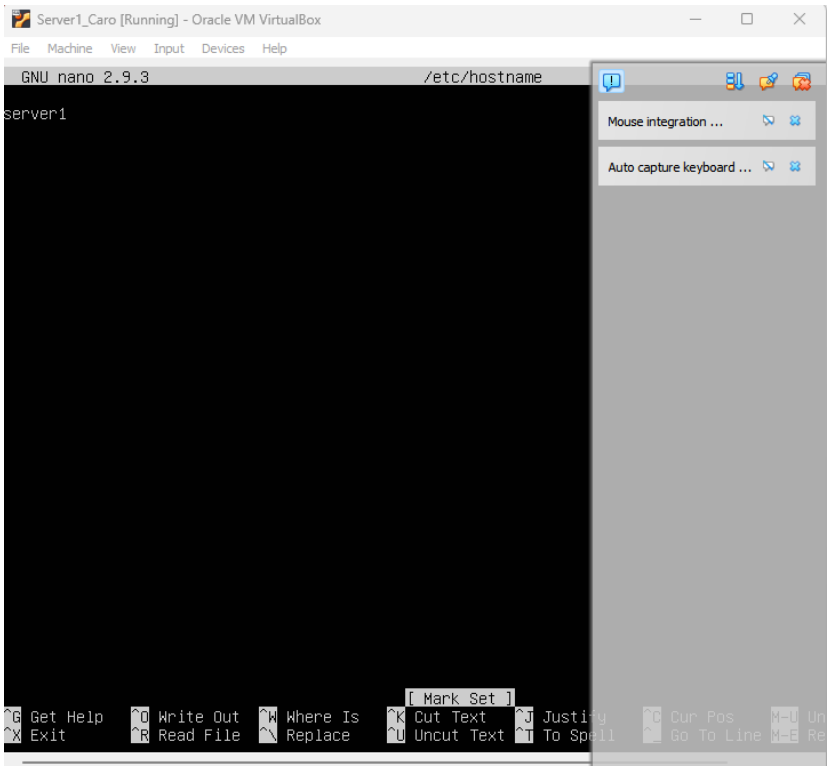
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



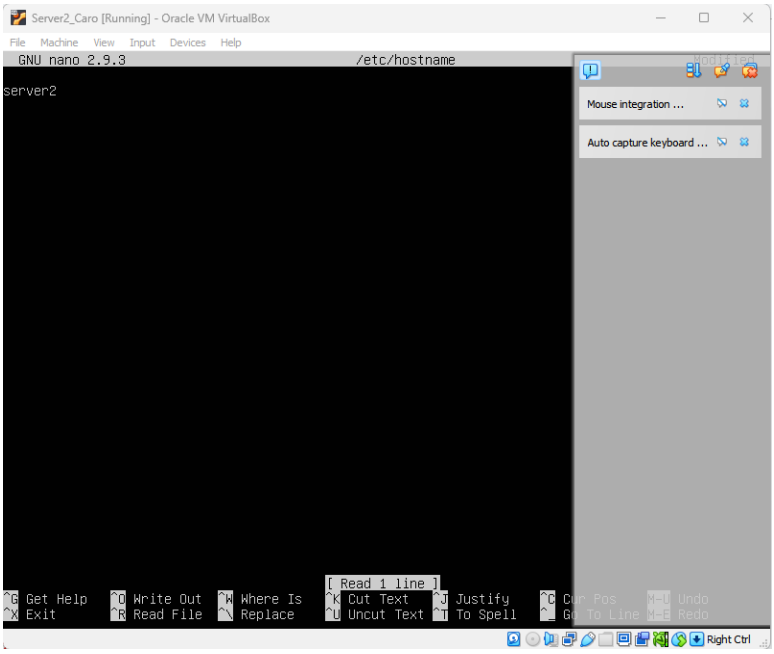
Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

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

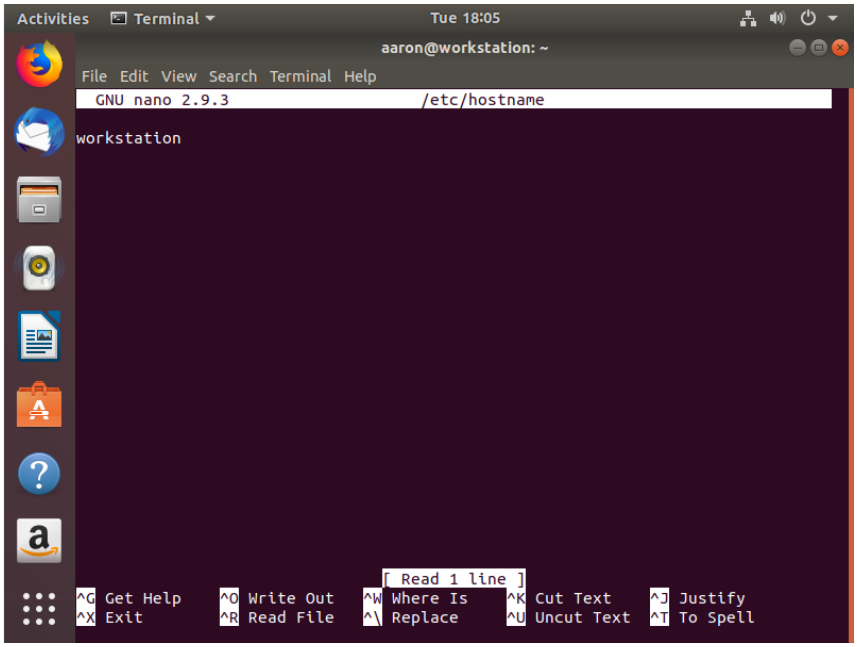
1.1Use server1 for Server 1



1.2Use server2 for Server 2

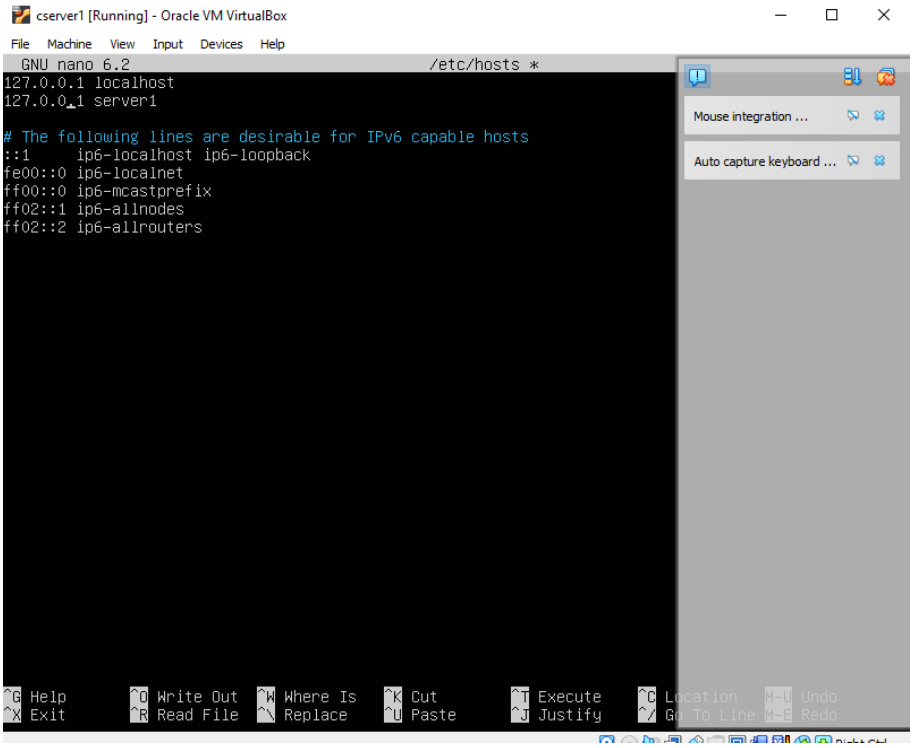


1.3Use workstation for the Local Machine

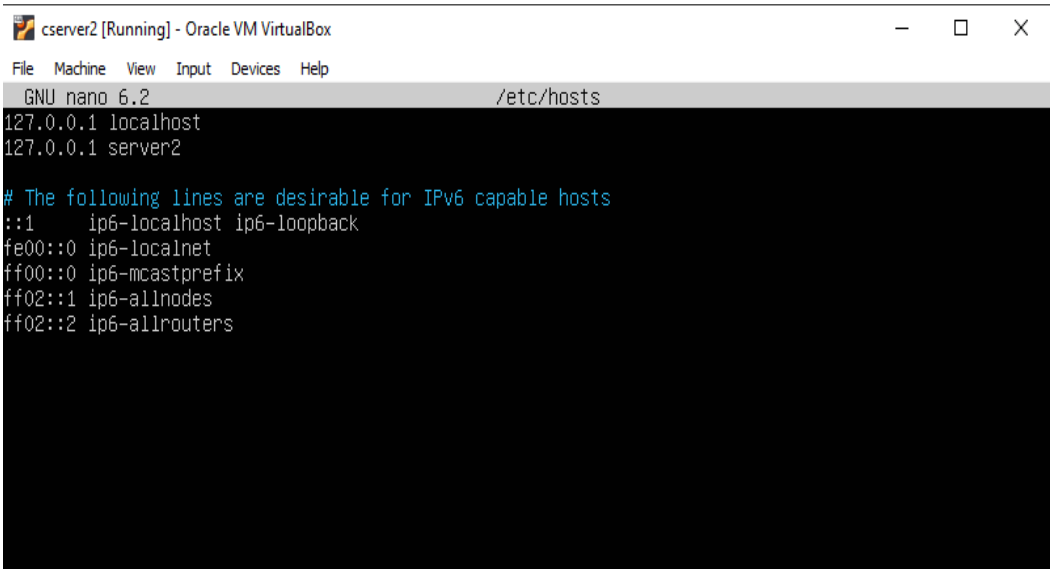


2. Edit the hosts using the command *sudo nano /etc/hosts.* Edit the second line.

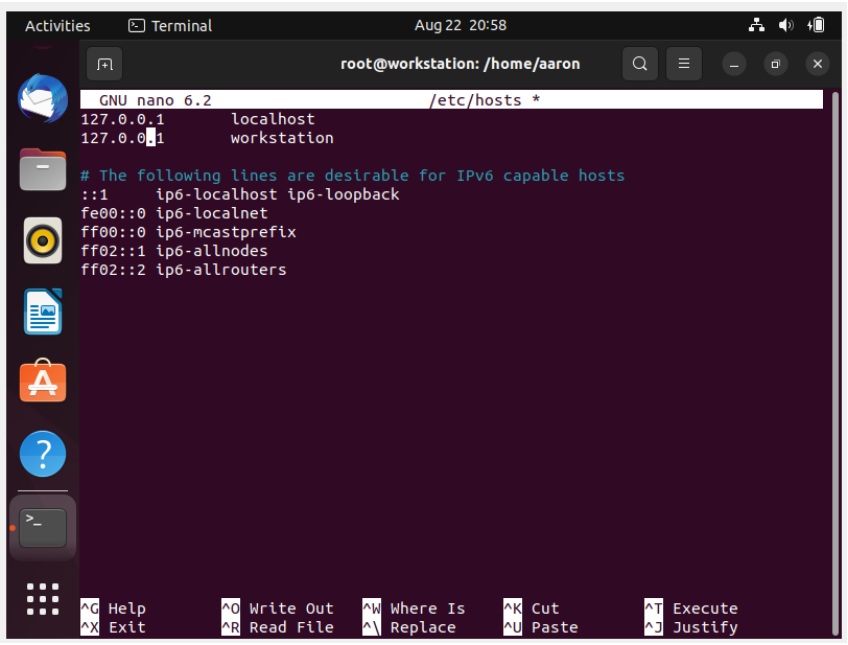
2.1Type 127.0.0.1 server 1 for Server 1



2.2Type 127.0.0.1 server 2 for Server 2



2.3Type 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.

```
root@server1:/home/aaron# sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
root@server1:/home/aaron# sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils cloud-init git git-man initramfs-tools initramfs-tools-bin initramfs-tools-core
  libapt-pkg6.0 libldap-2.5-0 libldap-common sosreport
12 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,747 kB of archives.
After this operation, 838 kB disk space will be freed.
Do you want to continue? [Y/n] _
```

```
root@server2:/home/aaron# sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
root@server2:/home/aaron# sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils cloud-init git git-man initramfs-tools initramfs-tools-bin initramfs-tools-core
  libapt-pkg6.0 libldap-2.5-0 libldap-common sosreport
12 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,747 kB of archives.
After this operation, 838 kB disk space will be freed.
Do you want to continue? [Y/n]
```

```
root@workstation:/home/aaron# sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 110 kB in 2s (67.4 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
26 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
root@workstation:/home/aaron# sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  gjs libgjs0g
The following packages will be upgraded:
  apt apt-utils ghostscript ghostscript-x gir1.2-javascriptcoregtk-4.0
  gir1.2-webkit2-4.0 initramfs-tools initramfs-tools-bin initramfs-tools-core
  intel-microcode libapt-pkg6.0 libgs9 libgs9-common
  libjavascriptcoregtk-4.0-18 libldap-2.5-0 libldap-common libsmbclient
  libtiff5 libwbclient0 libwebkit2gtk-4.0-37 samba-libs vim-common vim-tiny
  xxd
```

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

```
root@server1:/home/aaron# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
root@server2:/home/aaron# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
Activities   Terminal   Aug 22 21:23
root@workstation: /home/aaron
root@workstation:/home/aaron# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
```

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*

```
root@server1:/home/aaron# sudo service ssh start
root@server1:/home/aaron# 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 Tue 2023-08-22 12:34:06 UTC; 58min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 729 (sshd)
      Tasks: 1 (limit: 4556)
     Memory: 4.4M
        CPU: 43ms
     CGroup: /system.slice/ssh.service
             └─729 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 12:34:06 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 12:34:06 server1 sshd[729]: Server listening on 0.0.0.0 port 22.
Aug 22 12:34:06 server1 sshd[729]: Server listening on :: port 22.
Aug 22 12:34:06 server1 systemd[1]: Started OpenBSD Secure Shell server.
```

```
root@server2:/home/aaron# sudo service ssh start
root@server2:/home/aaron# 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 Tue 2023-08-22 12:49:06 UTC; 45min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 709 (sshd)
      Tasks: 1 (limit: 4557)
     Memory: 4.4M
        CPU: 45ms
     CGroup: /system.slice/ssh.service
             └─709 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 12:49:06 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 12:49:06 server2 sshd[709]: Server listening on 0.0.0.0 port 22.
Aug 22 12:49:06 server2 sshd[709]: Server listening on :: port 22.
Aug 22 12:49:06 server2 systemd[1]: Started OpenBSD Secure Shell server.
```

```
root@workstation:/home/aaron# sudo service ssh start
root@workstation:/home/aaron# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Tue 2023-08-22 21:22:58 +08; 7min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 9895 (sshd)
      Tasks: 1 (limit: 4591)
     Memory: 1.7M
        CPU: 52ms
     CGroup: /system.slice/ssh.service
             └─9895 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 21:22:58 workstation systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 21:22:58 workstation sshd[9895]: Server listening on 0.0.0.0 port 22.
Aug 22 21:22:58 workstation sshd[9895]: Server listening on :: port 22.
Aug 22 21:22:58 workstation systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (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*

```
root@server1:/home/aaron# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@server1:/home/aaron# sudo ufw enable
Firewall is active and enabled on system startup
root@server1:/home/aaron# sudo ufw status
Status: active

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

```
root@server2:/home/aaron# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@server2:/home/aaron# sudo ufw enable
Firewall is active and enabled on system startup
root@server2:/home/aaron# sudo ufw status
Status: active

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

```
root@workstation:/home/aaron# sudo ufw allow ssh
Rules updated
Rules updated (v6)
root@workstation:/home/aaron# sudo ufw enable
Firewall is active and enabled on system startup
root@workstation:/home/aaron# sudo ufw status
Status: active

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

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.

   1.1Server 1 IP address: 192.168.56.102

   1.2Server 2 IP address: 192.168.56.103

1.3Workstation IP address: 192.168.56.101

```
aaron@server1:~$ ifconfig
enp0s3: 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::a00:27ff:fe97:51d3  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:97:51:d3  txqueuelen 1000  (Ethernet)
        RX packets 16  bytes 6065 (6.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 1474 (1.4 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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 560  bytes 40000 (40.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 560  bytes 40000 (40.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
aaron@server2:~$ ifconfig
enp0s3: 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::a00:27ff:fe28:9d69  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:28:9d:69  txqueuelen 1000  (Ethernet)
        RX packets 16  bytes 6065 (6.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1544 (1.5 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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 720  bytes 51360 (51.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 720  bytes 51360 (51.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
aaron@workstation:~$ ifconfig
enp0s3: 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::556e:ef93:884a:d341  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1e:41:27  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 5546 (5.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61  bytes 8376 (8.3 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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 639  bytes 47419 (47.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 639  bytes 47419 (47.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Make sure that they can ping each other.

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

```
aaron@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=0.970 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.462 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.651 ms
^Z
[1]+  Stopped                 ping 192.168.56.102
aaron@workstation:~$
```

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

```
aaron@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=0.685 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.534 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.557 ms
^Z
[2]+  Stopped                 ping 192.168.56.103
```

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

```
aaron@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=0.993 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.593 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.549 ms
^Z
[1]+  Stopped                 ping 192.168.56.103
```

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*

1.2Enter the password for server 1 when prompted

```
aaron@workstation:~$ ssh aaron@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established
.
ED25519 key fingerprint is SHA256:G9vvQ9jDGxQTCyKsB2NAhFunO9OhpJBgCiUAT7QWcIE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known host
s.
aaron@192.168.56.102's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 02:37:48 PM UTC 2023

  System load:  0.080078125        Processes:               124
  Usage of /:   44.5% of 11.21GB   Users logged in:         1
  Memory usage: 6%                 IPv4 address for enp0s3: 192.168.56.102
  Swap usage:   0%
```

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.*

```
aaron@server1:~$ aaron@server1
aaron@server1: command not found
aaron@server1:~$
logout
Connection to 192.168.56.102 closed.
aaron@workstation:~$
```

3. Do the same for Server 2.

```
aaron@workstation:~$ ssh aaron@192.168.56.103 -y
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established
.
ED25519 key fingerprint is SHA256:vTa5VHNmIfZGBzBPOdXXI4aOuerbT5oSbqje63uYJGE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
aaron@192.168.56.103's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 02:41:22 PM UTC 2023

  System load:  0.0                Processes:               113
  Usage of /:   44.5% of 11.21GB   Users logged in:         1
  Memory usage: 5%                 IPv4 address for enp0s3: 192.168.56.103
  Swap usage:   0%
```

```
Last login: Tue Aug 22 14:25:31 2023
aaron@server2:~$
logout
Connection to 192.168.56.103 closed.
aaron@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)

4.3Save the file and exit.

```
Activities   Terminal   Aug 22 22:49
root@workstation: /home/aaron
  GNU nano 6.2                    /etc/hosts *
127.0.0.1       localhost
127.0.0.1       workstation
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
```

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.

```
aaron@workstation:~$ ssh aaron@server1
The authenticity of host 'server1 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:G9vvQ9jDGxQTCyKsB2NAhFunO9OhpJBgCiUAT7QWcIE.
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 'server1' (ED25519) to the list of known hosts.
aaron@server1's password:
Permission denied, please try again.
aaron@server1's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)
```

```
aaron@workstation:~$ ssh aaron@server2
The authenticity of host 'server2 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:vTa5VHNmIfZGBzBPOdXXI4aOuerbT5oSbqje63uYJGE.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:4: [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.
aaron@server2's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

  System information as of Tue Aug 22 02:52:25 PM UTC 2023

  System load:  0.0                Processes:               112
  Usage of /:   44.5% of 11.21GB   Users logged in:         1
  Memory usage: 5%                 IPv4 address for enp0s3: 192.168.56.103
  Swap usage:   0%
```

Reflections:

Answer the following:

1. **How are we able to use the hostname instead of IP address in SSH commands?**

We can use the hostname in SSH commands because it relies on DNS (Domain Name System) resolution to translate the hostname into its corresponding IP address for establishing the connection.

2. How secured is SSH?

SSH is considered highly secure due to its encryption and authentication mechanisms, making it a widely trusted protocol for secure remote access and data transfer.

Conclusion:

In conclusion, virtual networks play a pivotal role in modern computing environments by enabling the seamless integration of virtual machines (VMs) into network infrastructures. The ability to create, configure, and test the connectivity of VMs within these virtual networks is essential for ensuring the robustness and functionality of complex IT systems. By effectively managing and validating VM connectivity, organizations can optimize their network resources, enhance security, and streamline operations in today's dynamic digital landscape.