Name: Atienza, Stephen Lhaye C.	**Date Performed:01/16/24**
Course/Section:CPE231S1	**Date Submitted:01/16/24**
Instructor: Dr. Jonathan V. Taylar	**Semester and SY: 01/16/24**

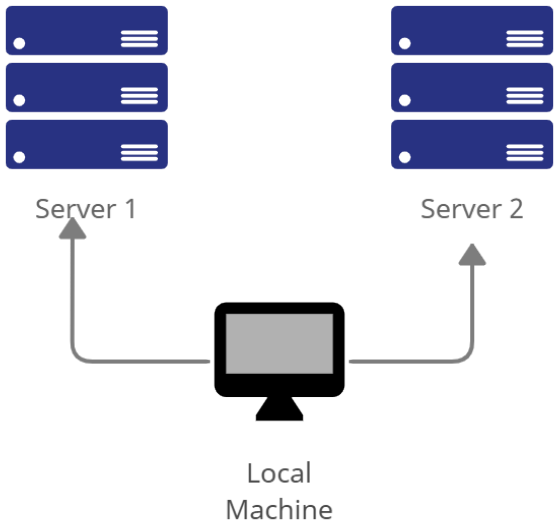
Activity 1: Configure Network using Virtual Machines

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:**

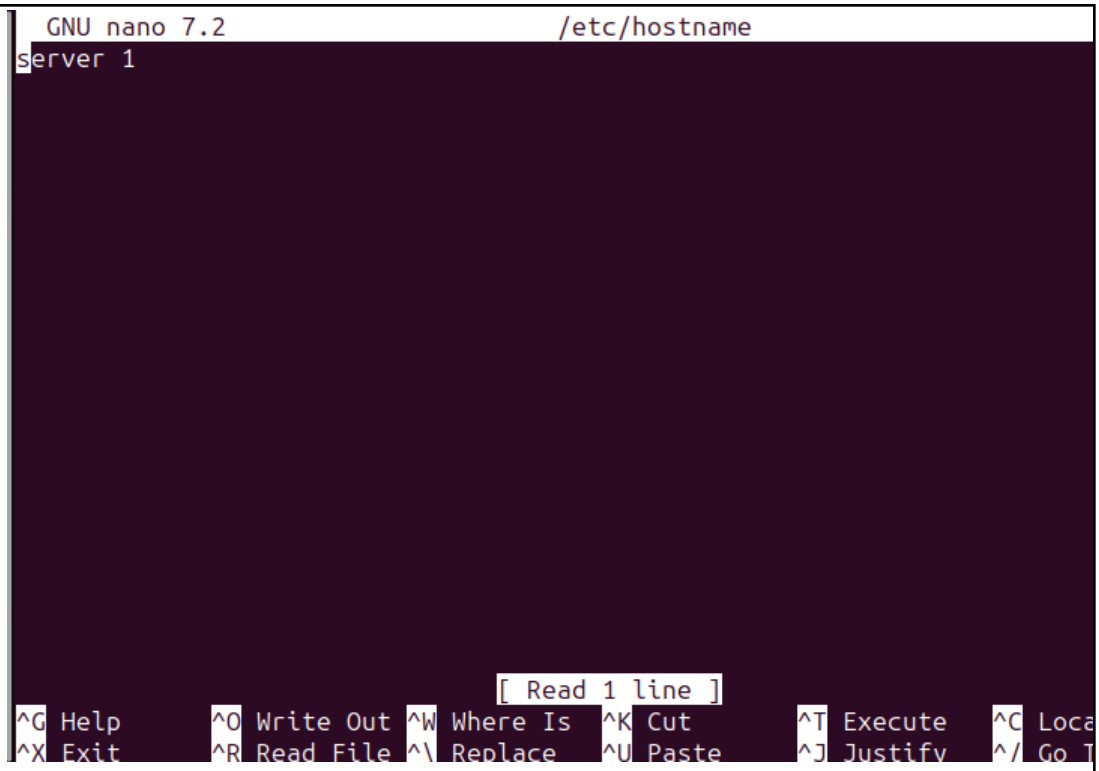
Network Topology:

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*).

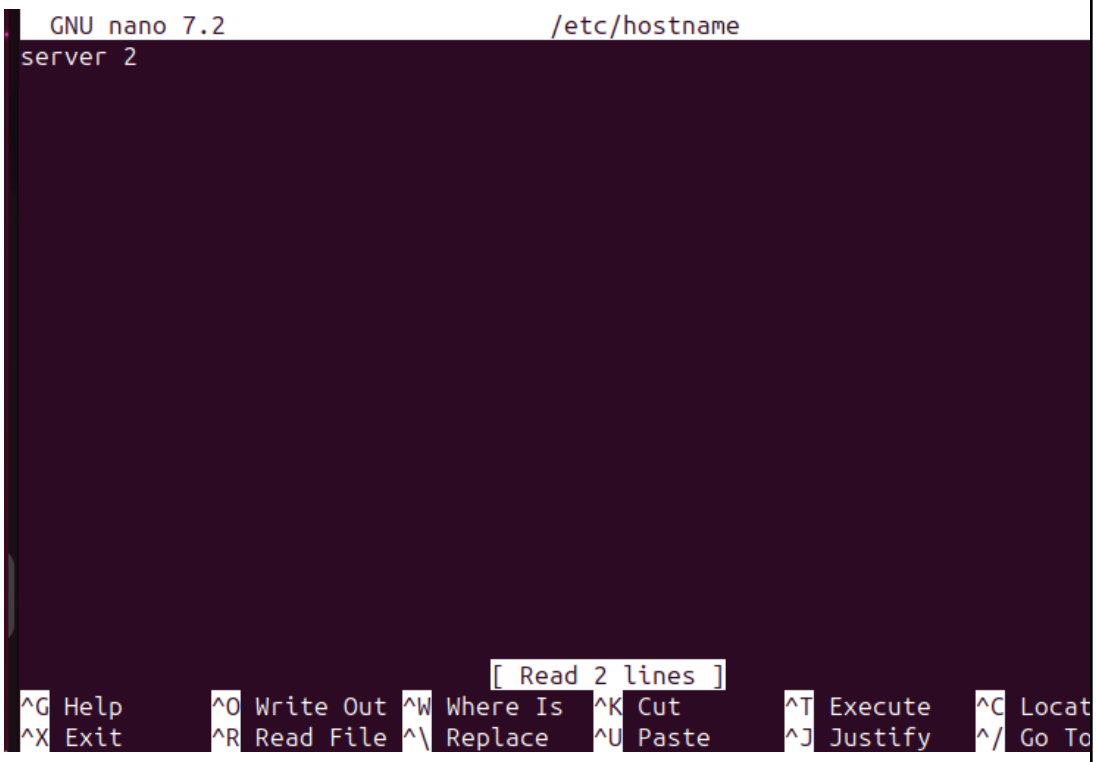


Task 1: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.

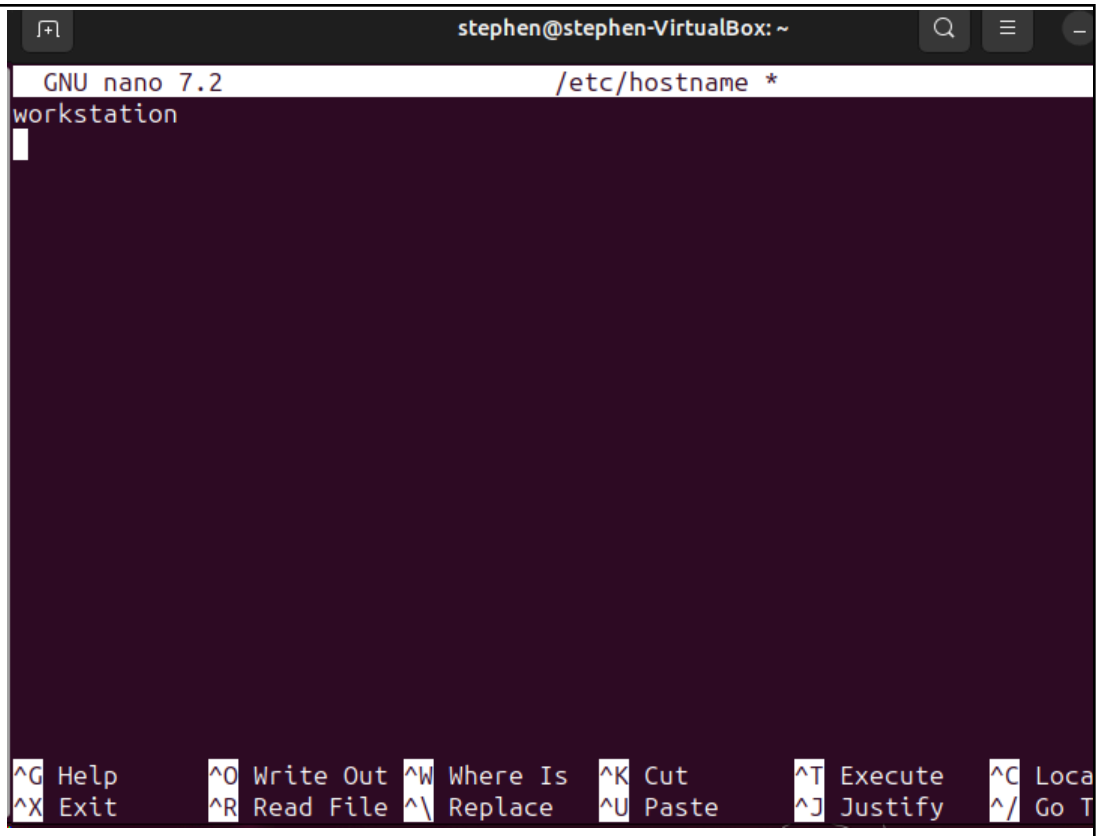
1. Change the hostname using the command *sudo nano /etc/hostname*
   1.1 Use server1 for Server 1



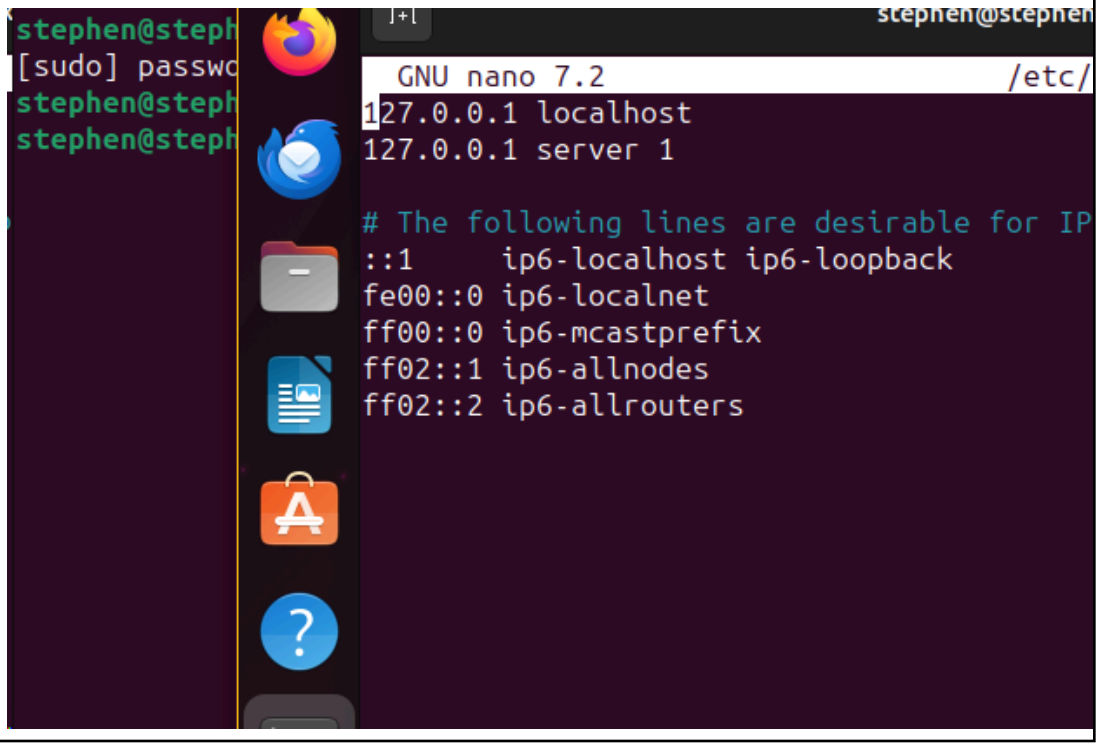
1.2 Use server2 for Server 2



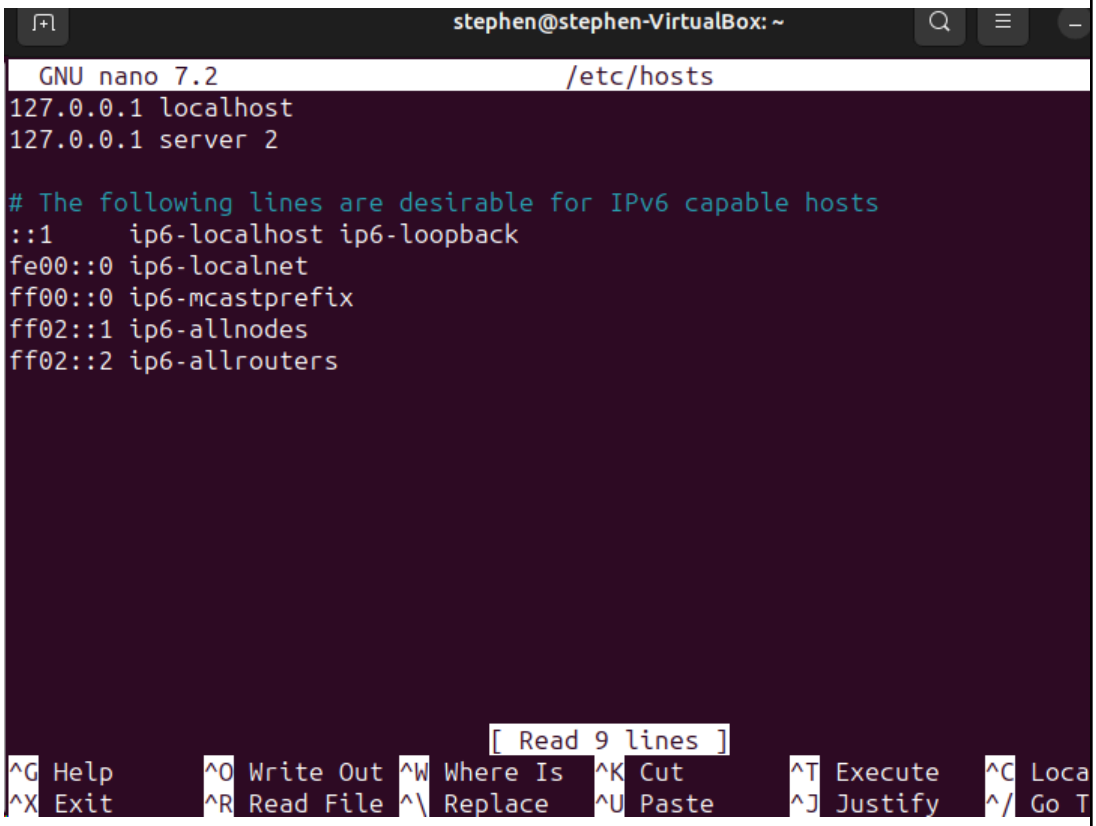
1.3 Use workstation for the Local Machine



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



2.2 Type 127.0.0.1 server 2 for Server 2



2.3 Type 127.0.0.1 workstation for the Local Machine

```
  GNU nano 7.2                          /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









                        [ Wrote 9 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Loca
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go T
```

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*.

```
stephen@stephen-VirtualBox:~$ sudo apt install openssh-server -y
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 129 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,103 kB of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu lunar-updates/main amd64 openssh
server amd64 1:9.0p1-1ubuntu8.7 [38.1 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu lunar-updates/main amd64 openssh
r amd64 1:9.0p1-1ubuntu8.7 [432 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu lunar-updates/main amd64 ncurses
all 6.4-2ubuntu0.1 [272 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu lunar/main amd64 ssh-import-id a
1-0ubuntu1 [10.1 kB]
Fetched 751 kB in 0s (1,874 kB/s)
Preconfiguring packages ...
```

3. Verify if the SSH service has started by issuing the following commands:
   3.1 *sudo service ssh start*

```
stephen@stephen-VirtualBox:~$ sudo service ssh start
```

   3.2 *sudo systemctl status ssh*

```
stephen@stephen-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset:
    Drop-In: /etc/systemd/system/ssh.service.d
             └─00-socket.conf
     Active: active (running) since Tue 2024-01-16 18:03:23 PST; 27s ag
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3186 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0
   Main PID: 3187 (sshd)
      Tasks: 1 (limit: 4485)
     Memory: 1.5M
        CPU: 13ms
     CGroup: /system.slice/ssh.service
             └─3187 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 sta

Jan 16 18:03:23 stephen-VirtualBox systemd[1]: Starting ssh.service - O
Jan 16 18:03:23 stephen-VirtualBox sshd[3187]: Server listening on :: p
Jan 16 18:03:23 stephen-VirtualBox systemd[1]: Started ssh.service - Op
lines 1-19/19 (END)
```

4. Configure the firewall to all port 22 by issuing the following commands:
   4.1 *sudo ufw allow ssh*

```
stephen@stephen-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

   4.2 *sudo ufw enable*

```
stephen@stephen-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

   4.3 *sudo ufw status*

```
stephen@stephen-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
stephen@stephen-VirtualBox:~$ 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.

1.1 Server 1 IP address: 192.168.56.___

```
stephen@stephen-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::6837:a289:fb1b:c59f  prefixlen 64  scopeid 0x20<link
        ether 08:00:27:d5:85:67  txqueuelen 1000  (Ethernet)
        RX packets 865  bytes 1020303 (1.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 524  bytes 45835 (45.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.109  netmask 255.255.255.0  broadcast 192.168.56
        inet6 fe80::e9b4:d376:a8bf:df42  prefixlen 64  scopeid 0x20<link
        ether 08:00:27:58:f2:bb  txqueuelen 1000  (Ethernet)
        RX packets 828  bytes 98986 (98.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 585  bytes 69026 (69.0 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)
```

1.2 Server 2 IP address: 192.168.56.___

```
stephen@stephen-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::c118:fa9c:29c1:1925  prefixlen 64  scopeid 0x20<l
        ether 08:00:27:f2:35:eb  txqueuelen 1000  (Ethernet)
        RX packets 866  bytes 1020323 (1.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 537  bytes 47045 (47.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.110  netmask 255.255.255.0  broadcast 192.168
        inet6 fe80::24b1:e240:4d82:96a1  prefixlen 64  scopeid 0x20<l
        ether 08:00:27:13:f2:2f  txqueuelen 1000  (Ethernet)
        RX packets 323  bytes 46784 (46.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 136  bytes 26290 (26.2 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)
```

1.3 Server 3 IP address: 192.168.56.___

```
stephen@workstation:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::22f2:324a:327c:6079  prefixlen 64  scopeid 0x20<lir
        ether 08:00:27:e3:a4:55  txqueuelen 1000  (Ethernet)
        RX packets 1103  bytes 1037203 (1.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 855  bytes 77705 (77.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.108  netmask 255.255.255.0  broadcast 192.168.5
        inet6 fe80::981c:22eb:3fb4:3965  prefixlen 64  scopeid 0x20<lir
        ether 08:00:27:52:3e:c8  txqueuelen 1000  (Ethernet)
        RX packets 873  bytes 106518 (106.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 576  bytes 67383 (67.3 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)
```

2. Make sure that they can ping each other.

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

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

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

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

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

```
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 '192.168.56.109' (ED25519) to the list of known
.
stephen@192.168.56.109's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

128 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


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.
```

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

```
stephen@server1:~$
logout
Connection to 192.168.56.109 closed.
stephen@workstation:~$
```

3. Do the same for Server 2.

```
Warning: Permanently added '192.168.56.110' (ED25519) to the list of known
.
stephen@192.168.56.110's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

128 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


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.

stephen@server2:~$
logout
Connection to 192.168.56.110 closed.
```

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)

```
  GNU nano 7.2                          /etc/hosts
127.0.0.1 localhost
127.0.0.1 workstation
192.168.56.109 server 1
192.168.56.110 server 2

# 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)

4.3 Save the file and exit.

```
  GNU nano 7.2                                /etc/hosts
127.0.0.1 localhost
127.0.0.1 workstation
192.168.56.109 server 1
192.168.56.110 server 2

# 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.

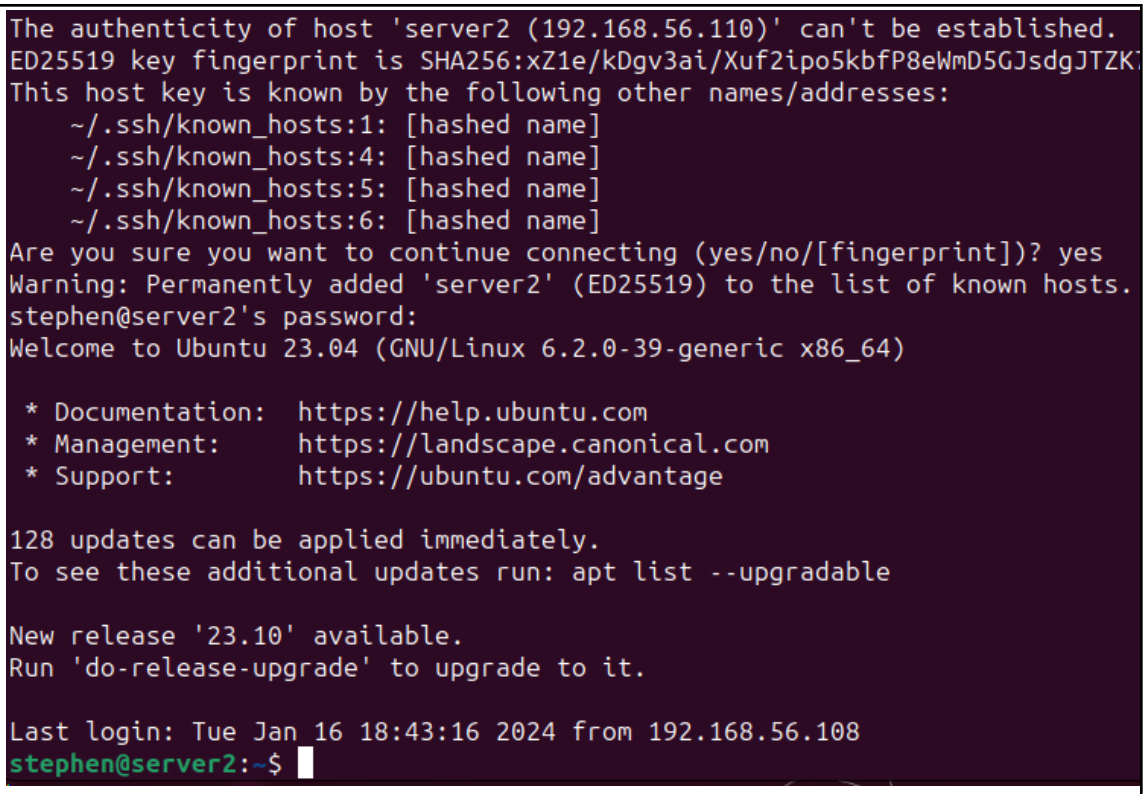
```
The authenticity of host 'server1 (192.168.56.109)' can't be established.
ED25519 key fingerprint is SHA256:xZ1e/kDgv3ai/Xuf2ipo5kbfP8eWmD5GJsdgJTZk
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: Yes
Warning: Permanently added 'server1' (ED25519) to the list of known hosts.
stephen@server1's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-39-generic x86_64)

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

128 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan 16 18:41:03 2024 from 192.168.56.108
stephen@server1:~$
```



Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
   - The IP address can be substituted for the hostname by appending the IP address and control nodes to the directory via sudo nano /etc/hostname.

2. How secured is SSH?
   - SSH ensures security by employing user authentication for any modifications and encrypting the keys.

Conclusion:

- Achieving the required goals demonstrates a thorough grasp of cloud computing and network design. You successfully established and fine-tuned Virtual Machines using Microsoft Azure or VirtualBox, demonstrating your ability to create and manage virtualized environments. Successfully establishing a Virtual Network and verifying connection demonstrates your ability to manage complex network infrastructures. Your achievements demonstrate your competence in virtualization, cloud solutions, and network administration, which are essential skills in the rapidly evolving digital landscape.