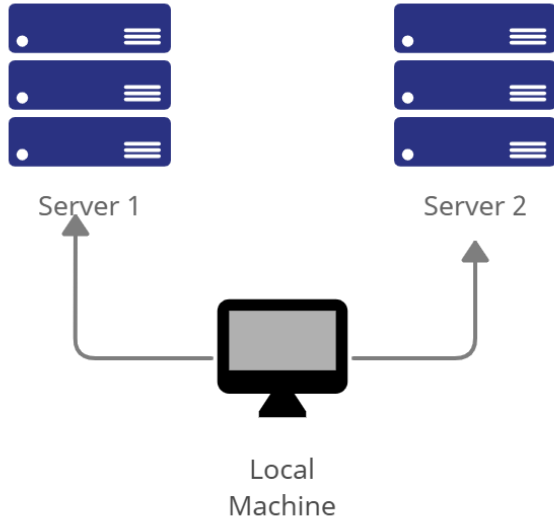


Name: Carlo Hideki D. Ilagan	Date Performed: 8/25/2024
Course/Section: CpE31S2	Date Submitted: 8/25/2024
Instructor: Sir. Robin Valenzuela	Semester and SY: 1st Sem/2024-2025
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft Azure 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, <i>provide screenshots for each task.</i> (Note: it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine).	
 <pre> graph TD LocalMachine[Local Machine] --> Server1[Server 1] LocalMachine --> Server2[Server 2] </pre> <p>The diagram illustrates a network topology. At the bottom center is a computer icon labeled "Local Machine". Two lines extend upwards from the "Local Machine" to two server racks. The left server rack is labeled "Server 1" and the right server rack is labeled "Server 2". Each server rack consists of three blue rectangular units stacked vertically, each with a small circle and horizontal lines on its front panel.</p>	
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 <i>sudo nano /etc/hostname</i> 1.1 Use server1 for Server 1	

```
hideki@hideki-VirtualBox: ~  
GNU nano 7.2 /etc/hostname  
server1
```

1.2 Use server2 for Server 2

```
GNU nano 7.2 /etc/hostname  
server2
```

1.3 Use workstation for the Local Machine

```
hideki@hideki-VirtualBox: ~  
GNU nano 7.2 /etc/hostname  
workstation
```

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

```
hideki@hideki-VirtualBox: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1 server1  
127.0.1.1 hideki-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  
  
[ Wrote 9 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
```

2.2 Type 127.0.0.1 server 2 for Server 2

```
hideki@hideki-VirtualBox: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1 server2  
127.0.1.1 hideki-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

```
hideki@hideki-VirtualBox: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1 workstation  
127.0.1.1 hideki-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  
  
[ Wrote 9 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
```

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.

```
hideki@hideki-VirtualBox:~$ sudo apt update  
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]  
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Package  
hideki@hideki-VirtualBox:~$ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

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

```
hideki@workstation: ~  
hideki@workstation:~$ sudo apt install openssh-server  
[sudo] password for hideki:  
Reading package lists... Done  
Building dependency tree... Done
```

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*

```
hideki@hideki-VirtualBox:~$ sudo service ssh start  
hideki@hideki-VirtualBox:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>  
   Active: active (running) since Sun 2024-08-25 10:58:13 PST; 9s ago  
TriggeredBy: ● ssh.socket  
   Docs: man:sshd(8)  
         man:sshd_config(5)  
  Process: 24580 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS>  
 Main PID: 24582 (sshd)  
    Tasks: 1 (limit: 4616)  
  Memory: 1.2M (peak: 1.5M)  
     CPU: 21ms  
   CGroup: /system.slice/ssh.service  
           └─24582 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Aug 25 10:58:13 hideki-VirtualBox systemd[1]: Starting ssh.service - OpenBSD Se>  
Aug 25 10:58:13 hideki-VirtualBox sshd[24582]: Server listening on :: port 22.  
Aug 25 10:58:13 hideki-VirtualBox systemd[1]: Started ssh.service - OpenBSD Sec>  
lines 1-17/17 (END)
```

```

hideki@workstation:~$ sudo service ssh start
hideki@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 10:58:47 PST; 6s ago
 TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 3175 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 3176 (sshd)
    Tasks: 1 (limit: 4616)
   Memory: 1.2M (peak: 1.5M)
      CPU: 21ms
   CGroup: /system.slice/ssh.service
           └─3176 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 10:58:47 workstation systemd[1]: Starting ssh.service - OpenBSD Secure S
Aug 25 10:58:47 workstation sshd[3176]: Server listening on :: port 22.
Aug 25 10:58:47 workstation systemd[1]: Started ssh.service - OpenBSD Secure Sh
lines 1-17/17 (END)

```

```

hideki@hideki-VirtualBox:~$ sudo service ssh start
hideki@hideki-VirtualBox:~$ 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 11:27:58 PST; 7s ago
 TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 24632 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 24634 (sshd)
    Tasks: 1 (limit: 4616)
   Memory: 1.2M (peak: 1.5M)
      CPU: 23ms
   CGroup: /system.slice/ssh.service
           └─24634 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 25 11:27:58 hideki-VirtualBox systemd[1]: Starting ssh.service - OpenBSD Se
Aug 25 11:27:58 hideki-VirtualBox sshd[24634]: Server listening on :: port 22.
Aug 25 11:27:58 hideki-VirtualBox systemd[1]: Started ssh.service - OpenBSD Sec
lines 1-17/17 (END)
[1]+  Stopped                  sudo systemctl status ssh
hideki@hideki-VirtualBox:~$

```

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*

```
hideki@hideki-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
hideki@hideki-VirtualBox:~$ sudo ufw enable
sFirewall is active and enabled on system startup
hideki@hideki-VirtualBox:~$ sudo ufw status
Status: active
```

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

```
hideki@hideki-VirtualBox:~$
```

```
hideki@hideki-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
hideki@hideki-VirtualBox:~$ sudo ufw enable
suFirewall is active and enabled on system startup
hideki@hideki-VirtualBox:~$ sudo ufw status
Status: active
```

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

```
hideki@hideki-VirtualBox:~$
```

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

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

```
hideki@workstation:~$
```

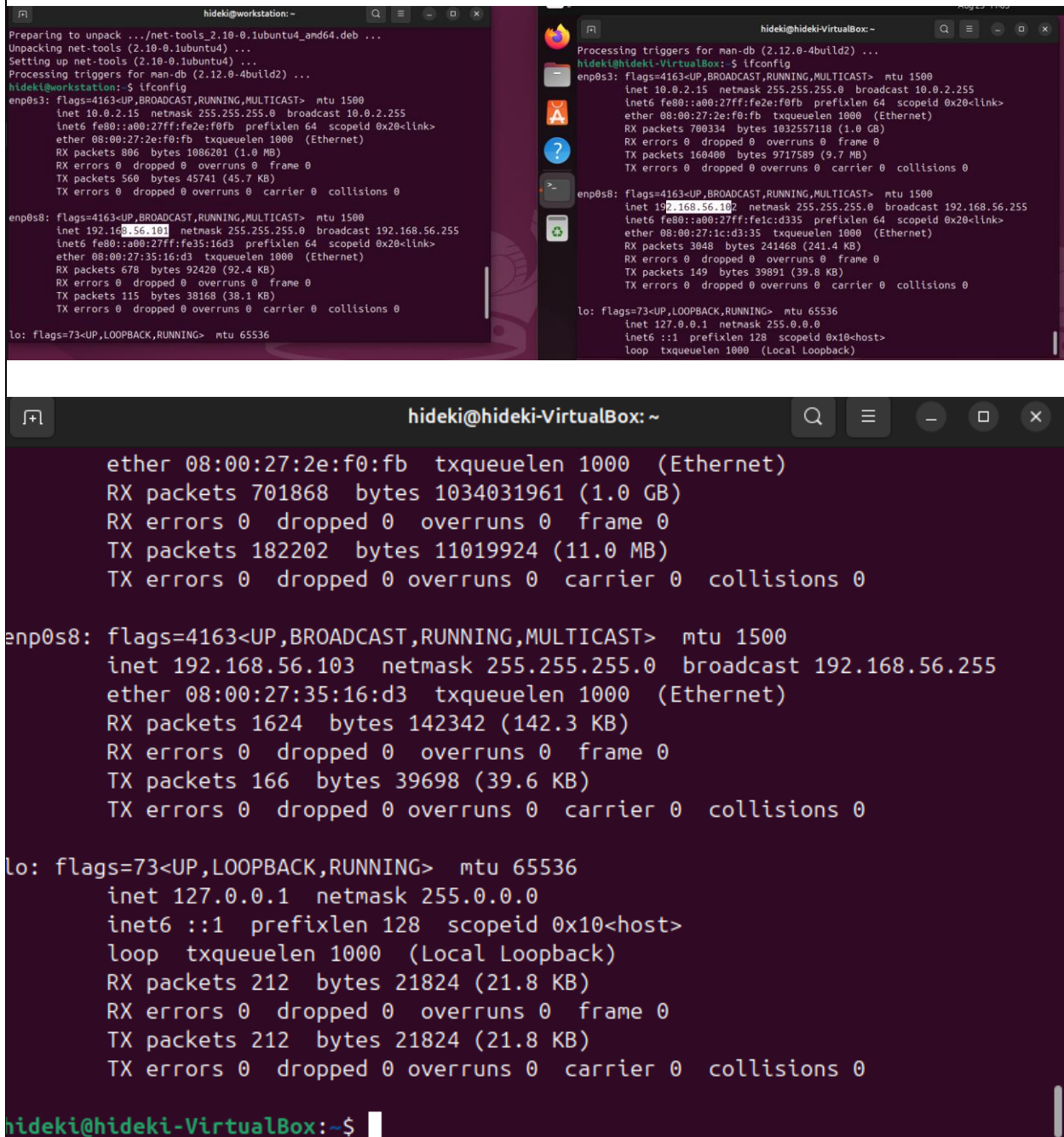
Task 3: Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:

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 workstation IP address: 192.168.56.101



```
hideki@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::a00:27ff:fe2e:f0fb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:2e:f0:fb txqueuelen 1000 (Ethernet)
    RX packets 806 bytes 1086201 (1.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 560 bytes 45741 (45.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe35:16d3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:35:16:d3 txqueuelen 1000 (Ethernet)
    RX packets 678 bytes 92420 (92.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 115 bytes 38168 (38.1 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 212 bytes 21824 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 212 bytes 21824 (21.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hideki@hideki-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::a00:27ff:fe2e:f0fb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:2e:f0:fb txqueuelen 1000 (Ethernet)
    RX packets 700334 bytes 1032557118 (1.0 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 160400 bytes 9717589 (9.7 MB)
    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::a00:27ff:fe1c:d335 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1c:d3:35 txqueuelen 1000 (Ethernet)
    RX packets 3048 bytes 241468 (241.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 149 bytes 39891 (39.8 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 212 bytes 21824 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 212 bytes 21824 (21.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hideki@hideki-VirtualBox:~$ 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
    ether 08:00:27:35:16:d3 txqueuelen 1000 (Ethernet)
    RX packets 1624 bytes 142342 (142.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 166 bytes 39698 (39.6 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 212 bytes 21824 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 212 bytes 21824 (21.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hideki@hideki-VirtualBox:~$
```

2. Make sure that they can ping each other.

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

```
hideki@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=0.679 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.355 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.319 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.352 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.244 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.349 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.306 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.266 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.262 ms
```

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

```
hideki@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=0.644 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.392 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.260 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.273 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.413 ms
```

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

```
hideki@hideki-VirtualBox:~$ 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=0.629 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.387 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.305 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`

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`

```
hideki@workstation:~$ ssh hideki@192.168.56.102
hideki@192.168.56.102's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-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

*** System restart required ***

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
```

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

```
hideki@hideki-VirtualBox:~$
logout
Connection to 192.168.56.102 closed.
```

3. Do the same for Server 2.

```
hideki@workstation:~$ ssh hideki@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:dM+cAdyhWK1u3RGiWaFbRheWMUtl4kc5451ehLY4Fao.
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 '192.168.56.103' (ED25519) to the list of known hosts
.
hideki@192.168.56.103's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)

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

```
hideki@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)
- 4.3 Save the file and exit.

```
GNU nano 7.2 /etc/hosts
127.0.0.1 workstation
127.0.1.1 hideki-VirtualBox
192.168.56.102 server1
192.168.56.103 server2
```

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.

```
Last login: Sun Aug 25 11:12:42 2024 from 192.168.56.101
hideki@server1:~$
```

```
Last login: Sun Aug 25 11:38:27 2024 from 192.168.56.101
hideki@server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
 - By adding the two IP addresses of the servers inside the /etc/hosts file of the workstation PC.
2. How secured is SSH?
 - The SSH is very secured since it is asking for the computer password or the machine password to be able to access it rather than being accessed without any authentication required.