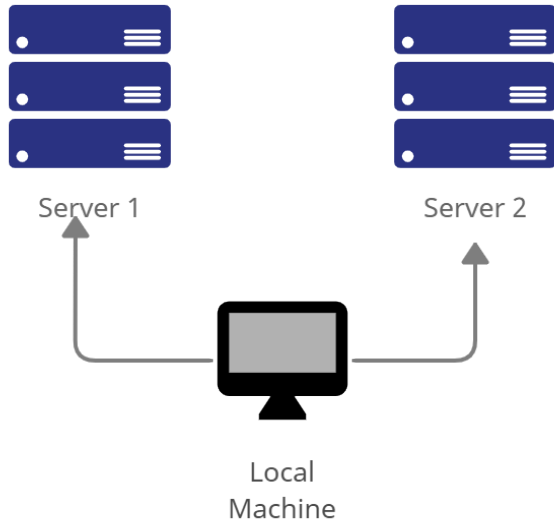
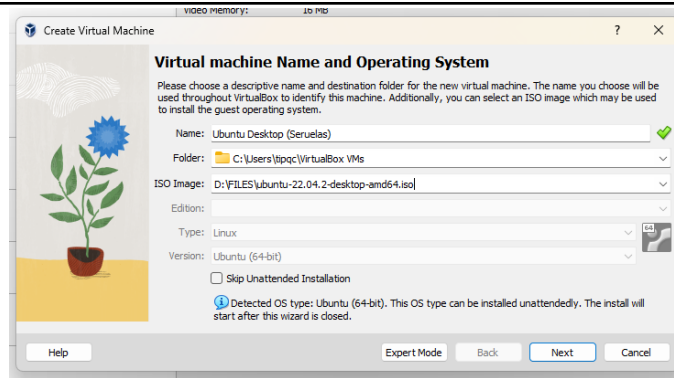
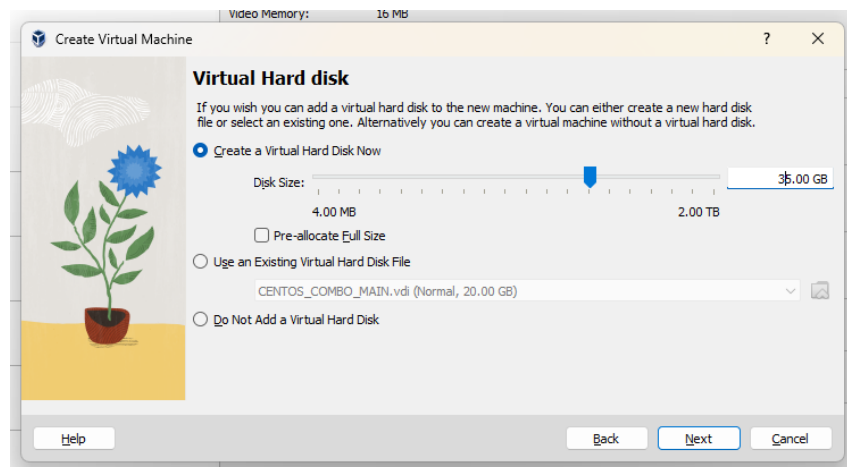
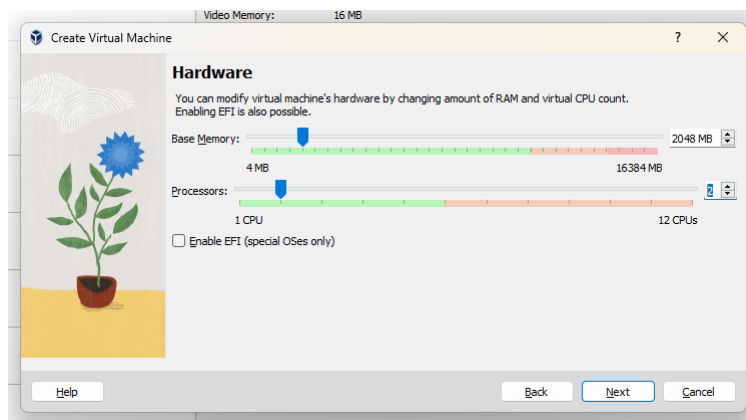


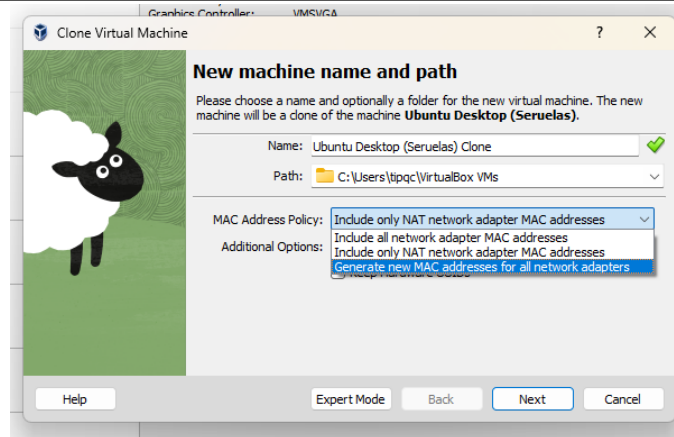
<b>Name:</b> Seruelas, Ronn Kristoper H.	<b>Date Performed:</b> 08/14/23
<b>Course/Section:</b> CPE31S4	<b>Date Submitted:</b> 08/15/23
<b>Instructor:</b> Dr. Jonathan V. Taylor	<b>Semester and SY:</b> 1st Sem. 2023-2024
<b>Activity 1: Configure Network using Virtual Machines</b>	
<b>1. Objectives:</b> 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 1.2. Set-up a Virtual Network and Test Connectivity of VMs	
<b>2. Discussion:</b>  <b>Network Topology:</b> Assume that you have created the following network topology in Virtual Machines, <i>provide screenshots for each task.</i> (Note: <i>it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine</i> ).	
 <pre> graph TD     LocalMachine[Local Machine] --&gt; Server1[Server 1]     LocalMachine --&gt; Server2[Server 2] </pre> <p>The diagram illustrates a network topology where a central 'Local Machine' (represented by a monitor icon) is connected to two separate server stacks. 'Server 1' on the left and 'Server 2' on the right each consist of three stacked server rack icons. Arrows point from the Local Machine to each of the two server stacks, indicating network connectivity.</p>	



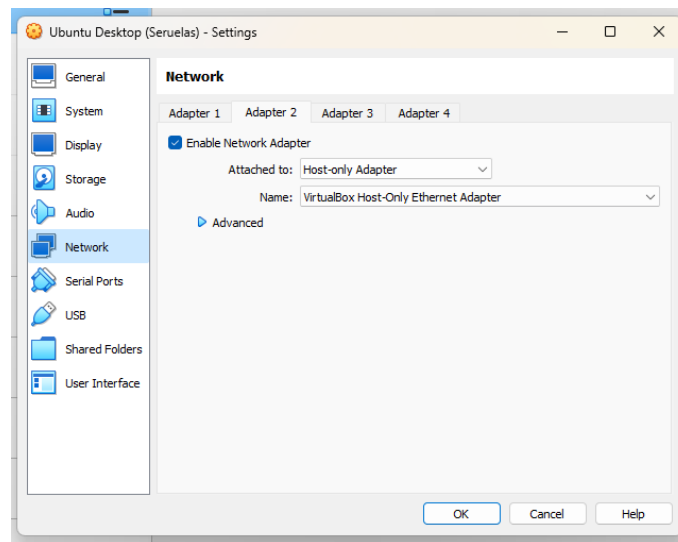
**Figure 1.1 - Creating a Virtual Machine for Ubuntu Linux.**



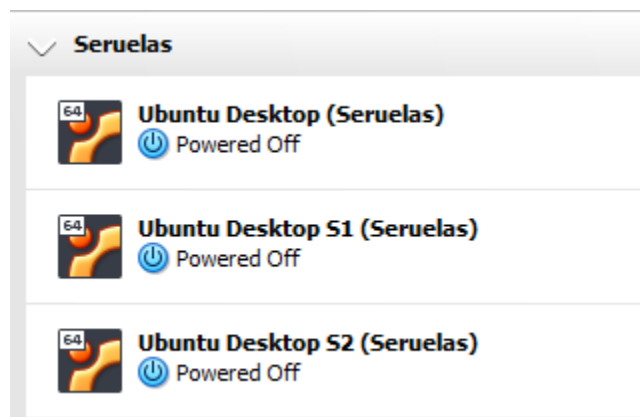
**Figure 1.2-1.3 - Allocating the resources for the Virtual Machine**



**Figure 1.4 - Cloning the Virtual Machine for 2 servers.**



**Figure 1.5 - Configuring Network Adapter #2 Settings to Host-Only Adapter, VirtualBox Host-Only Ethernet Adapter #2.**




**Figure 1.6 - Grouping all Virtual Machines together.**

**Task 2:** 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*

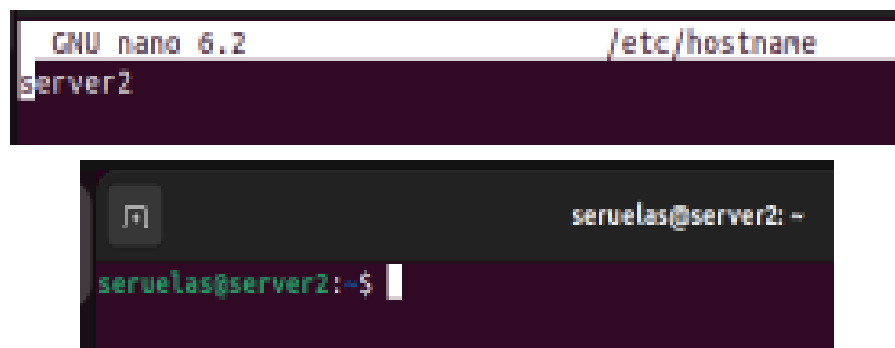
1.1 Use server1 for Server 1



```
GNU nano 6.2 /etc/hostname *
server1
```

```
seruelas@server1: ~
seruelas@server1:~$
```

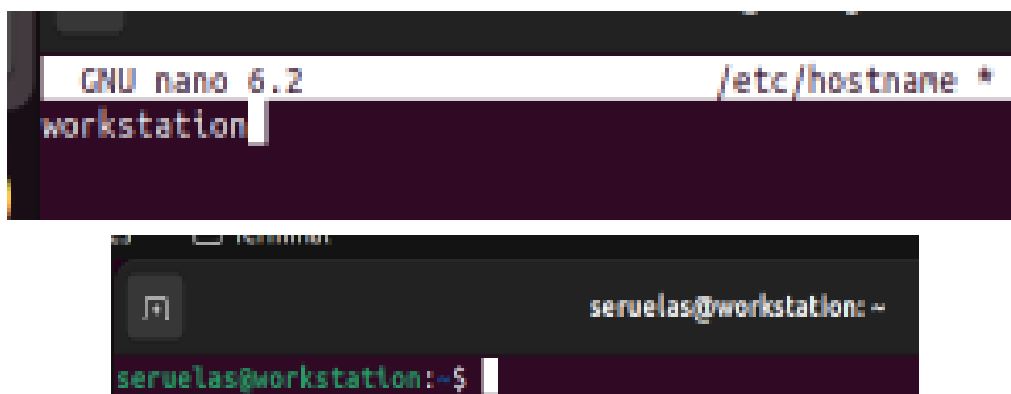
1.2 Use server2 for Server 2



```
GNU nano 6.2 /etc/hostname
server2
```

```
seruelas@server2: ~
seruelas@server2:~$
```

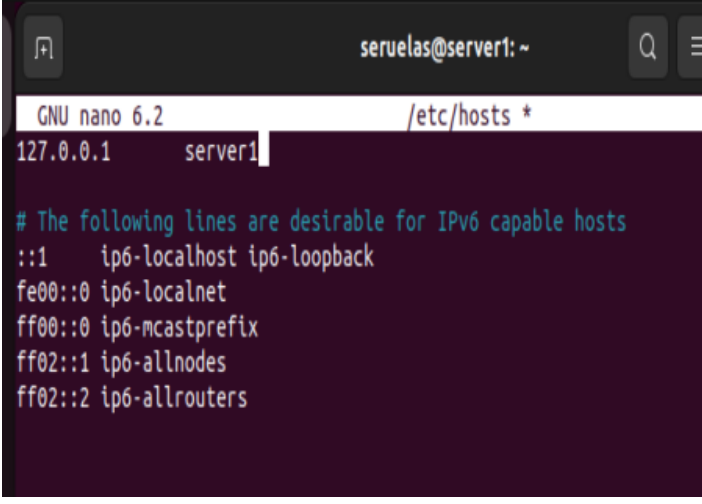
1.3 Use workstation for the Local Machine



```
GNU nano 6.2 /etc/hostname *
workstation
```

```
seruelas@workstation: ~
seruelas@workstation:~$
```

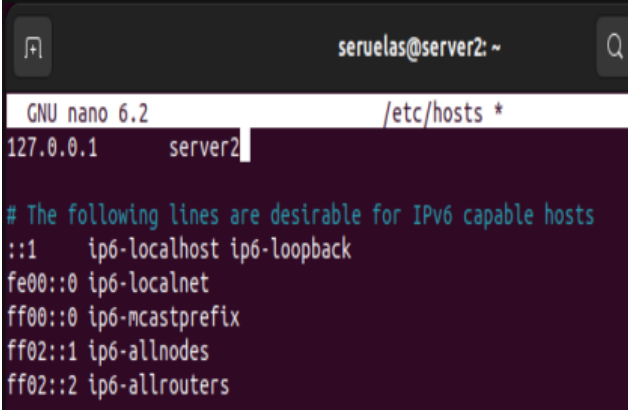
2. Edit the hosts using the command `sudo nano /etc/hosts`. Edit the second line.  
2.1 Type 127.0.0.1 server 1 for Server 1

A terminal window titled 'seruelas@server1: ~' showing the nano 6.2 editor editing /etc/hosts. The first line is '127.0.0.1 server1'. Below it are several lines of IPv6 configuration comments and addresses.

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 server1

# 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
```

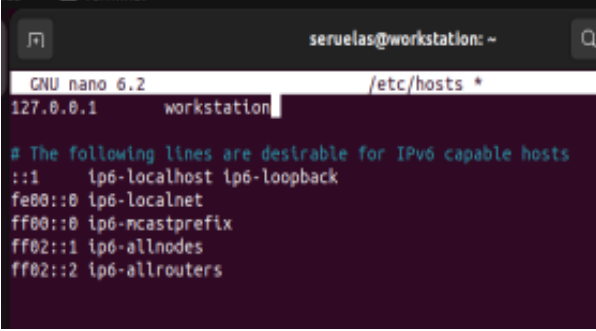
- 2.2 Type 127.0.0.1 server 2 for Server 2

A terminal window titled 'seruelas@server2: ~' showing the nano 6.2 editor editing /etc/hosts. The first line is '127.0.0.1 server2'. Below it are several lines of IPv6 configuration comments and addresses.

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 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
```

- 2.3 Type 127.0.0.1 workstation for the Local Machine

A terminal window titled 'seruelas@workstation: ~' showing the nano 6.2 editor editing /etc/hosts. The first line is '127.0.0.1 workstation'. Below it are several lines of IPv6 configuration comments and addresses.

```
GNU nano 6.2 /etc/hosts *
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 3:** 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.

```
seruelas@seruelas-ManagedNode:~$ sudo apt update | sudo apt upgrade
[sudo] password for seruelas:

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
```

**Figure 3.1** - Updating/Upgrading the machines prior cloning for more efficiency.

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

```
seruelas@workstation:~$ 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.
```

```
seruelas@server1:~$ sudo apt install openssh-server
[sudo] password for seruelas:
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.
```

```
seruelas@server2:~$ sudo apt install openssh-server
[sudo] password for seruelas:
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.
```

**Figure 3.2.1-3.2.3** - Installation of OpenSSH in Workstation, Server1, and Server2.

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*

```
seruelas@workstation: ~  
seruelas@workstation:~$ sudo service ssh start  
seruelas@workstation:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en  
   Active: active (running) since Tue 2023-08-15 17:04:23 PST; 1min 5s ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 2577 (sshd)  
     Tasks: 1 (limit: 2253)  
    Memory: 1.7M  
       CPU: 18ms  
    CGroup: /system.slice/ssh.service  
            └─2577 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Aug 15 17:04:23 workstation systemd[1]: Starting OpenBSD Secure Shell server...  
Aug 15 17:04:23 workstation sshd[2577]: Server listening on 0.0.0.0 port 22.  
Aug 15 17:04:23 workstation sshd[2577]: Server listening on :: port 22.  
Aug 15 17:04:23 workstation systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-16/16 (END)
```

```
seruelas@server1: ~  
seruelas@server1:~$ sudo service ssh start  
suseruelas@server1:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en  
   Active: active (running) since Tue 2023-08-15 17:04:46 PST; 1min 56s ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 2276 (sshd)  
     Tasks: 1 (limit: 2253)  
    Memory: 1.9M  
       CPU: 15ms  
    CGroup: /system.slice/ssh.service  
            └─2276 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Aug 15 17:04:46 server1 systemd[1]: Starting OpenBSD Secure Shell server...  
Aug 15 17:04:46 server1 sshd[2276]: Server listening on 0.0.0.0 port 22.  
Aug 15 17:04:46 server1 sshd[2276]: Server listening on :: port 22.  
Aug 15 17:04:46 server1 systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-16/16 (END)
```

```
seruelas@server2: ~  
seruelas@server2:~$ sudo service ssh start  
suseruelas@server2:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en  
   Active: active (running) since Tue 2023-08-15 17:06:16 PST; 56s ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 2207 (sshd)  
     Tasks: 1 (limit: 2253)  
    Memory: 1.7M  
       CPU: 15ms  
    CGroup: /system.slice/ssh.service  
            └─2207 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Aug 15 17:06:16 server2 systemd[1]: Starting OpenBSD Secure Shell server...  
Aug 15 17:06:16 server2 sshd[2207]: Server listening on 0.0.0.0 port 22.  
Aug 15 17:06:16 server2 sshd[2207]: Server listening on :: port 22.  
Aug 15 17:06:16 server2 systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-16/16 (END)
```

Figure 3.3.1-3.3.3 - Verifying the SSH Services done by executing the commands.

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*

```
seruelas@workstation: ~  
seruelas@workstation:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)  
seruelas@workstation:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
seruelas@workstation:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)
```

```
seruelas@server2: ~  
seruelas@server2:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)  
seruelas@server2:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
seruelas@server2:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)
```

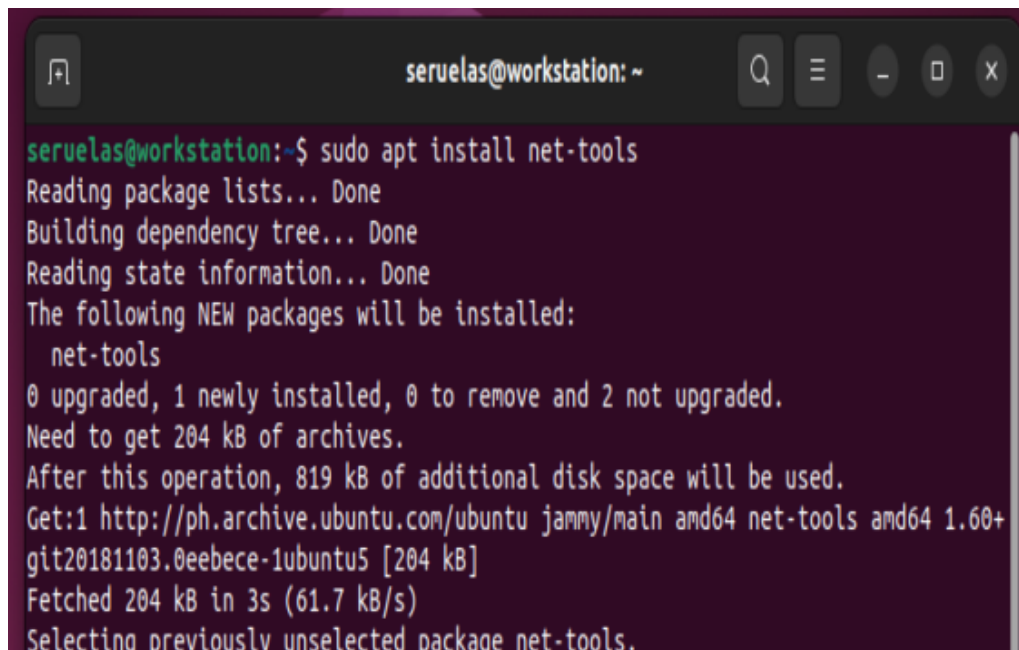
```
seruelas@server1: ~  
seruelas@server1:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)  
seruelas@server1:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
seruelas@server1:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)
```

Figure 3.4.1-3.4.3 - Configuring the firewall to all port 22.



**Task 4:** 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: 192.168.56.112
  - 1.2 Server 2 IP address: 192.168.56.113
  - 1.3 Local Machine IP address: 192.168.56.111

A terminal window titled 'seruelas@workstation: ~' with standard window controls. The terminal shows the command 'sudo apt install net-tools' being executed. The output indicates that the package 'net-tools' is being installed, requiring 204 kB of archives and 819 kB of additional disk space. The source is identified as 'http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]'. The download is complete, and the package is selected for installation.

```
seruelas@workstation:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+
git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 3s (61.7 kB/s)
Selecting previously unselected package net-tools.
```

**Figure 4.1.1** - Installation of package **net-tools** for the command **ifconfig**.

```
seruelas@server1:~$ 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::c3d8:b813:d4eb:e906 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b9:b7:08 txqueuelen 1000 (Ethernet)
    RX packets 28 bytes 5037 (5.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 102 bytes 11401 (11.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.112 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::e7dc:5a29:568d:2117 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e6:06:99 txqueuelen 1000 (Ethernet)
    RX packets 110 bytes 14179 (14.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 55 bytes 7133 (7.1 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 153 bytes 15148 (15.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 153 bytes 15148 (15.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

seruelas@server2:~$ 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::b98e:bcb0:918e:3a4b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:7d:ff:fa txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 5622 (5.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 106 bytes 11770 (11.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.113 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::975f:b019:8d4f:8f36 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:3d:38:30 txqueuelen 1000 (Ethernet)
    RX packets 68 bytes 9031 (9.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 68 bytes 8398 (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 147 bytes 14352 (14.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 14352 (14.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

seruelas@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::eb58:73c3:d313:15f1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:5f:81:77 txqueuelen 1000 (Ethernet)
    RX packets 37 bytes 6360 (6.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 113 bytes 12372 (12.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.111 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::5dc0:8370:49c:eabc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ed:d2:b9 txqueuelen 1000 (Ethernet)
    RX packets 93 bytes 12700 (12.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75 bytes 9027 (9.0 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)
    RX packets 154 bytes 15254 (15.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 154 bytes 15254 (15.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Figure 4.1.2-4.1.5 - Checking the IP Addresses of each machine.**

2. Make sure that they can ping each other.

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

```
seruelas@workstation: ~  
seruelas@workstation:~$ ping 192.168.56.112  
PING 192.168.56.112 (192.168.56.112) 56(84) bytes of data.  
64 bytes from 192.168.56.112: icmp_seq=1 ttl=64 time=1.21 ms  
64 bytes from 192.168.56.112: icmp_seq=2 ttl=64 time=0.974 ms  
64 bytes from 192.168.56.112: icmp_seq=3 ttl=64 time=0.792 ms  
^C  
--- 192.168.56.112 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 0.792/0.990/1.205/0.169 ms
```

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

```
seruelas@workstation: ~  
seruelas@workstation:~$ ping 192.168.56.113  
PING 192.168.56.113 (192.168.56.113) 56(84) bytes of data.  
64 bytes from 192.168.56.113: icmp_seq=1 ttl=64 time=0.874 ms  
64 bytes from 192.168.56.113: icmp_seq=2 ttl=64 time=0.894 ms  
64 bytes from 192.168.56.113: icmp_seq=3 ttl=64 time=1.44 ms  
^C  
--- 192.168.56.113 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2041ms  
rtt min/avg/max/mdev = 0.874/1.070/1.443/0.263 ms
```

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

```
seruelas@server1: ~  
seruelas@server1:~$ ping 192.168.56.113  
PING 192.168.56.113 (192.168.56.113) 56(84) bytes of data.  
64 bytes from 192.168.56.113: icmp_seq=1 ttl=64 time=1.51 ms  
64 bytes from 192.168.56.113: icmp_seq=2 ttl=64 time=0.447 ms  
64 bytes from 192.168.56.113: icmp_seq=3 ttl=64 time=1.69 ms  
^C  
--- 192.168.56.113 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2027ms  
rtt min/avg/max/mdev = 0.447/1.216/1.693/0.549 ms
```

**Task 5:** Verify SSH connectivity on Server 1, Server 2, and Local Machine.

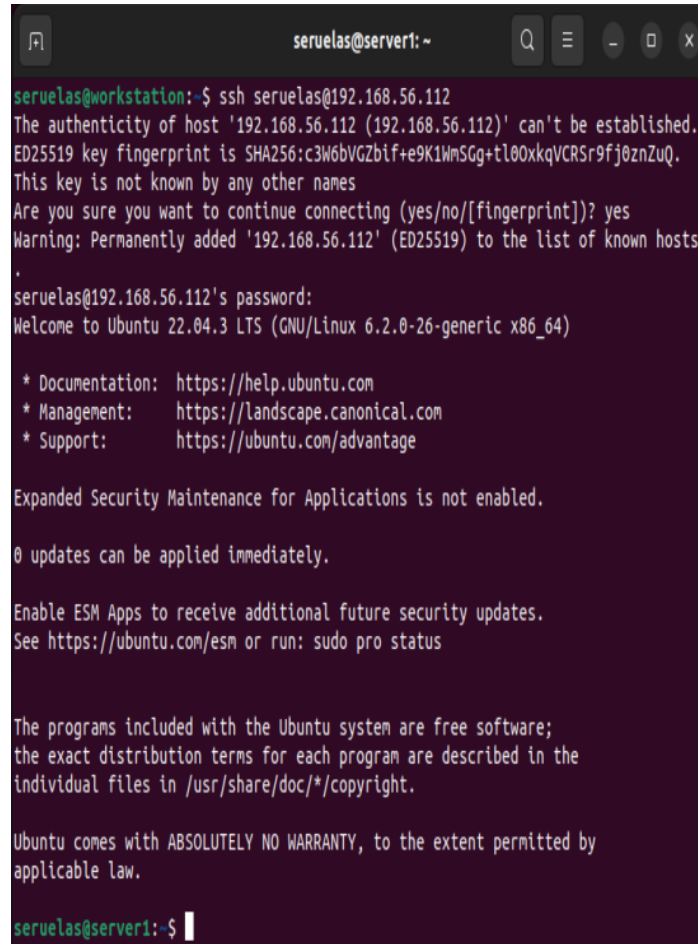
1. On the Local Machine, issue the following commands:

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`



```
seruelas@workstation:~$ ssh seruelas@192.168.56.112
The authenticity of host '192.168.56.112 (192.168.56.112)' can't be established.
ED25519 key fingerprint is SHA256:c3W6bVGZbif+e9K1WmSGg+tl00xkqVCRSr9fj0znZuQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.112' (ED25519) to the list of known hosts
seruelas@192.168.56.112's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)

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

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

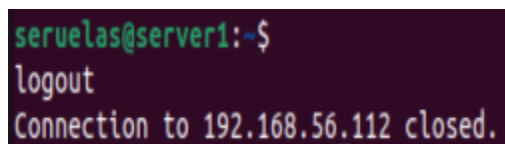
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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

seruelas@server1:~$
```

**Figure 5.1.1** - Remote hosting the Server 1 via SSH Server.

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



```
seruelas@server1:~$
logout
Connection to 192.168.56.112 closed.
```

**Figure 5.2.1** - Logging out of Server 1.

3. Do the same for Server 2.

A terminal window titled 'seruelas@workstation: ~' with standard window controls. The terminal shows an SSH session initiated from the workstation to a server at 192.168.56.113. It displays the host's fingerprint, a confirmation prompt, and the server's login banner for Ubuntu 22.04.3 LTS. The session ends with a logout message and a closed connection.

```
seruelas@workstation:~$ ssh seruelas@192.168.56.113
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established.
ED25519 key fingerprint is SHA256:cNWsEktjnqf3x+q3MgjSMafiAFEvHXunRC2J1y3oUK0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.113' (ED25519) to the list of known hosts
.
seruelas@192.168.56.113's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)

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

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

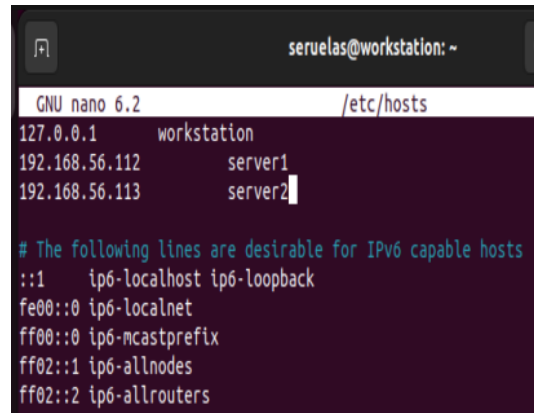
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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

seruelas@server2:~$
logout
Connection to 192.168.56.113 closed.
```

**Figure 5.3.1** - Remote hosting the Server 2 via SSH Server.

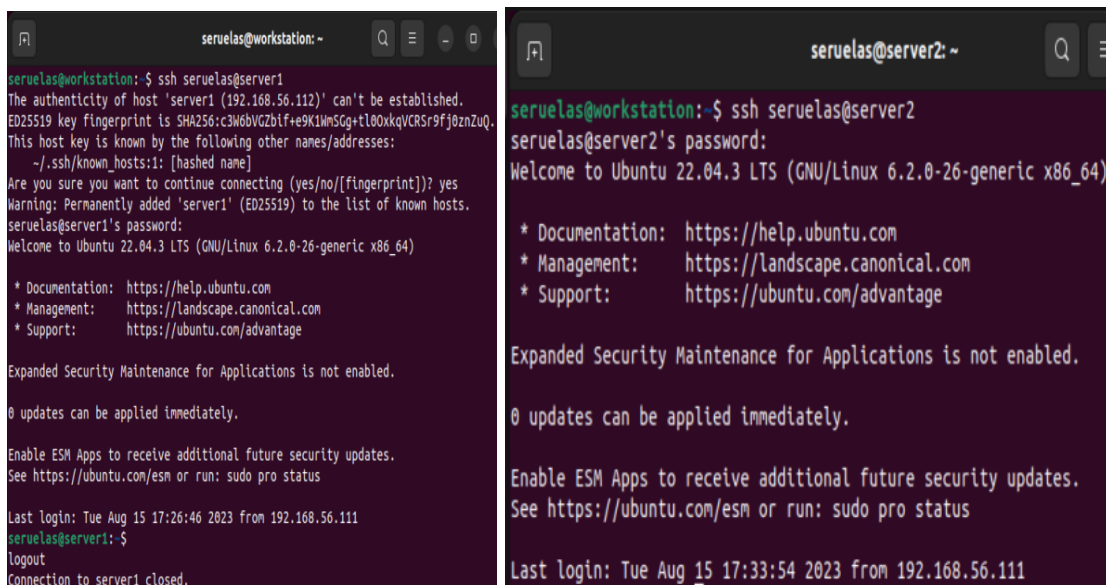
4. Edit the hosts of the Local Machine by issuing the command ***sudo nano /etc/hosts***. Below all texts type the following:
  - 4.1 **IP\_address server 1** (provide the ip address of server 1 followed by the hostname)
  - 4.2 **IP\_address server 2** (provide the ip address of server 2 followed by the hostname)
  - 4.3 Save the file and exit.



```
seruelas@workstation: ~  
GNU nano 6.2 /etc/hosts  
127.0.0.1    workstation  
192.168.56.112  server1  
192.168.56.113  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
```

**Figure 5.4.1** - Configuring the hosts by inputting their respective addresses.

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.



```
seruelas@workstation:~$ ssh seruelas@server1  
The authenticity of host 'server1 (192.168.56.112)' can't be established.  
ED25519 key fingerprint is SHA256:c3W6bVcZbif+e9K1WmSG+t100xkqVCRS9fj0znZuQ.  
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.  
seruelas@server1's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
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  
  
Last login: Tue Aug 15 17:26:46 2023 from 192.168.56.111  
seruelas@server1:~$  
logout  
Connection to server1 closed.  
  
seruelas@workstation:~$ ssh seruelas@server2  
seruelas@server2's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-26-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
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  
  
Last login: Tue Aug 15 17:33:54 2023 from 192.168.56.111
```

**Figure 5.5.1-5.5.2** - Verifying the hosts via SSH Command by using their hostnames.

**Reflections:**

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
  - It was made possible to use the hostnames of the IP addresses in the SSH commands by modifying or configuring the **/etc/hosts** file via **sudo nano**, by inputting the IP address of the server, then its hostname, allowing us to connect or remote host the server via their hostnames.
2. How secured is SSH?
  - SSH's security is more powerful and reliable as its security relies on encrypting its own traffic, using a public key that is used for authentication for its communications towards other computers.

***“I affirm that I have not received or given any unauthorized help on this activity and that all work is my own.”***