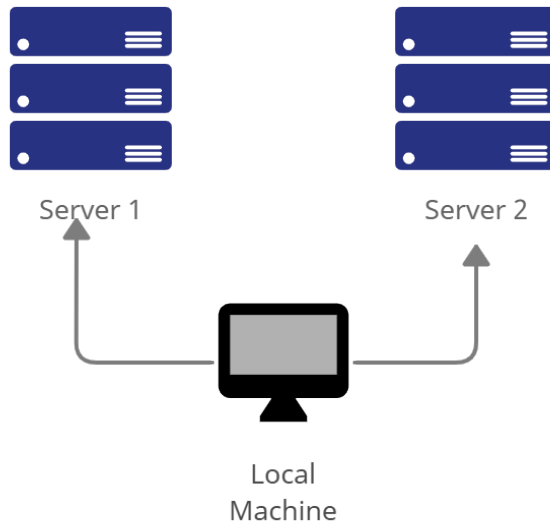


Name: Khelvin P. Nicolas	Date Performed: 8/22/2024
Course/Section: CPE 212 - CPE31S2	Date Submitted: 8/22/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem - 3rd Year
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft 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: <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] --> Server1[Server 1] LocalMachine --> 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 icons. Arrows point from the Local Machine to each of the two server stacks, indicating network connectivity.</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. <ol style="list-style-type: none"> 1. Change the hostname using the command <i>sudo nano /etc/hostname</i> <ol style="list-style-type: none"> 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

Task 1 Documentation:

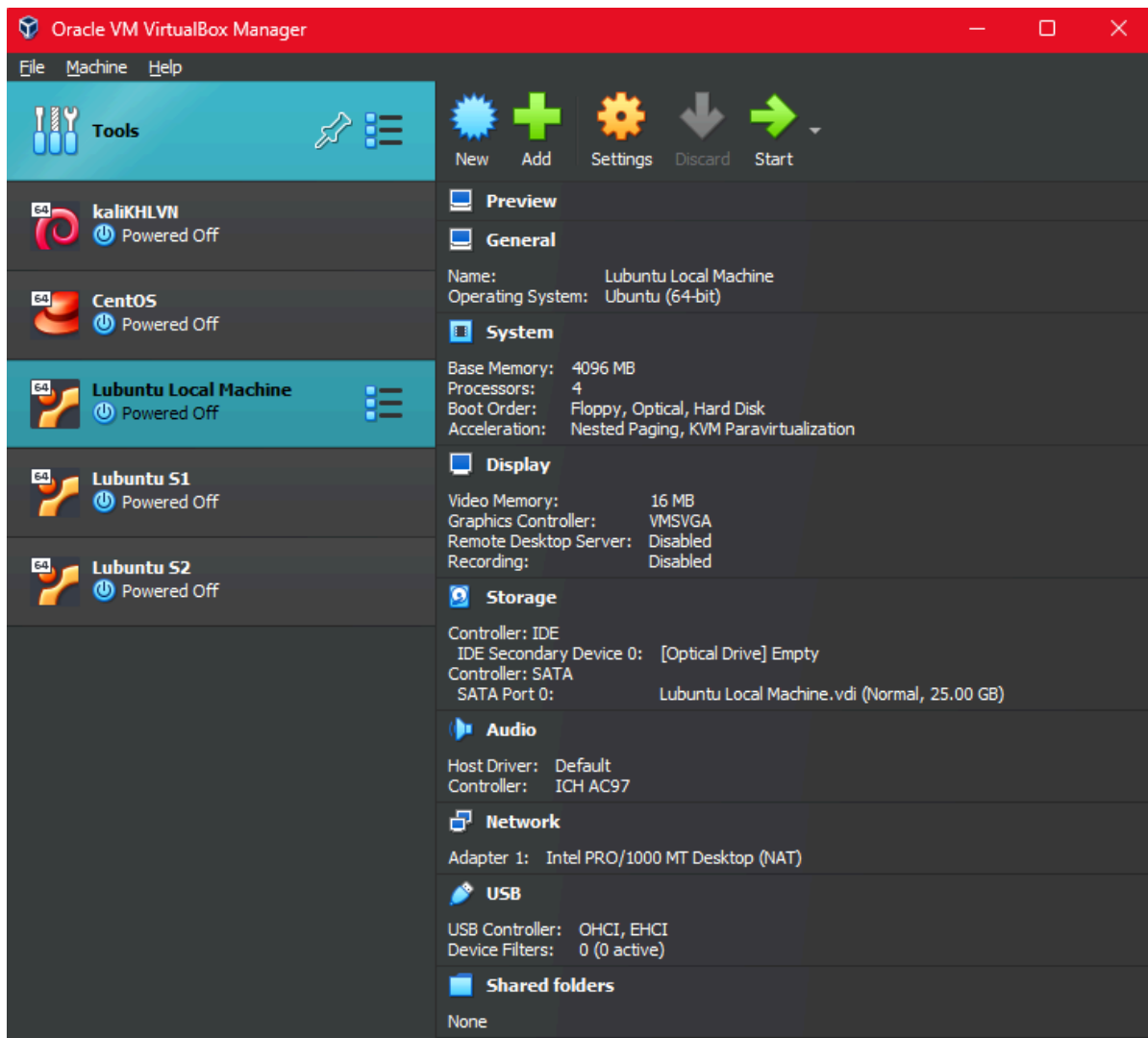


figure 1.1.1: VirtualBox VMs

```
khln@ubuntuKHLVN:~$ sudo nano /etc/hostname
[sudo] password for khln:
khln@ubuntuKHLVN:~$ cat /etc/hostname
server1
khln@ubuntuKHLVN:~$ █
```

figure 1.1.2: use server1 for Server 1

```
khln@ubuntuKHLVN:~$ sudo nano /etc/hostname
[sudo] password for khln:
khln@ubuntuKHLVN:~$ cat /etc/hostname
server2
khln@ubuntuKHLVN:~$ █
```

figure 1.1.3: use server2 for Server 2

```
khln@ubuntuKHLVN:~$ sudo nano /etc/hostname
[sudo] password for khln:
khln@ubuntuKHLVN:~$ cat /etc/hostname
workstation
khln@ubuntuKHLVN:~$ █
```

figure 1.1.4: use workstation for Local Machine

```
khln@server1:~$ cat /etc/hostname
server1
khln@server1:~$ sudo nano /etc/hosts
[sudo] password for khln:
khln@server1:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1    server1
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
# This host address
127.0.1.1    ubuntuKHLVN
khln@server1:~$ █
```

figure 1.2.1: Edit hosts for server1

```
khln@ubuntuKHLVN:~$ sudo nano /etc/hosts
khln@ubuntuKHLVN:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1    server2
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
# This host address
127.0.1.1    ubuntuKHLVN
khln@ubuntuKHLVN:~$ █
```

figure 1.2.2: Edit hosts for server2

```
khln@workstation:~$ sudo nano /etc/hosts
[sudo] password for khln:
khln@workstation:~$ cat/etc/hosts
bash: cat/etc/hosts: No such file or directory
khln@workstation:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1 workstation
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# This host address
127.0.1.1 ubuntuKHLVN
khln@workstation:~$ █
```

figure 1.2.3: Edit hosts for workstation

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*.
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*
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*

Task 2 Documentation:

```
khln@workstation:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1 workstation
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# This host address
127.0.1.1 ubuntuKHLVN
khln@workstation:~$ sudo apt update && sudo apt upgrade
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
]
```

```
khln@server1:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1 server1
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# This host address
127.0.1.1 ubuntuKHLVN
khln@server1:~$ sudo apt update && sudo apt upgrade
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
]
```

```
khln@ubuntuKHLVN:~$ cat /etc/hosts
# Standard host addresses
127.0.0.1 server2
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# This host address
127.0.1.1 ubuntuKHLVN
khln@ubuntuKHLVN:~$ sudo apt update && sudo apt upgrade
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
]
```

figure 2.1: apt update && apt upgrade on all VMs

```
khlvn@ubuntuKHLVN:~$ cat /etc/hostname
server2
khlvn@ubuntuKHLVN:~$ sudo apt install openssh-server
[sudo] password for khlvn:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
khlvn@server1:~$ cat /etc/hostname
server1
khlvn@server1:~$ sudo apt install openssh-server -y
[sudo] password for khlvn:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
khlvn@workstation:~$ cat /etc/hostname
workstation
khlvn@workstation:~$ sudo apt install openssh-server -y
[sudo] password for khlvn:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

figure 2.2: install openssh-server

```
khlvn@workstation:~$ sudo service ssh start
khlvn@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; p>
   Active: active (running) since Sun 2024-08-25 14:01:24 PST; 13s >
 TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
   Process: 18965 ExecStartPre=/usr/sbin/sshd -t (code=exited, statu>
 Main PID: 18966 (sshd)
    Tasks: 1 (limit: 4601)
   Memory: 1.2M (peak: 1.5M)
      CPU: 48ms
   CGroup: /system.slice/ssh.service
           └─18966 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 >

Ago 25 14:01:24 workstation systemd[1]: Starting ssh.service - OpenBS>
Ago 25 14:01:24 workstation sshd[18966]: Server listening on :: port >
Ago 25 14:01:24 workstation systemd[1]: Started ssh.service - OpenBSD>
khlvn@workstation:~$ █
```

```
khlvn@server1:~$ sudo service ssh start
khlvn@server1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; p>
   Active: active (running) since Sun 2024-08-25 14:03:02 PST; 6s a>
 TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
   Process: 18954 ExecStartPre=/usr/sbin/sshd -t (code=exited, statu>
 Main PID: 18955 (sshd)
    Tasks: 1 (limit: 4601)
   Memory: 1.2M (peak: 1.5M)
      CPU: 46ms
   CGroup: /system.slice/ssh.service
           └─18955 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 >

Ago 25 14:03:02 server1 systemd[1]: Starting ssh.service - OpenBSD Se>
Ago 25 14:03:02 server1 sshd[18955]: Server listening on :: port 22.
Ago 25 14:03:02 server1 systemd[1]: Started ssh.service - OpenBSD Sec>
khlvn@server1:~$ █
```



```

khlvn@ubuntuKHLVN:~$ sudo service ssh start
khlvn@ubuntuKHLVN:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; p>
   Active: active (running) since Sun 2024-08-25 14:03:42 PST; 18s >
 TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
   Process: 18974 ExecStartPre=/usr/sbin/sshd -t (code=exited, statu>
  Main PID: 18976 (sshd)
    Tasks: 1 (limit: 4601)
   Memory: 1.2M (peak: 1.4M)
      CPU: 49ms
   CGroup: /system.slice/ssh.service
           └─18976 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 >

Ago 25 14:03:41 ubuntuKHLVN systemd[1]: Starting ssh.service - OpenBS>
Ago 25 14:03:42 ubuntuKHLVN sshd[18976]: Server listening on :: port >
Ago 25 14:03:42 ubuntuKHLVN systemd[1]: Started ssh.service - OpenBSD>

```

figure 2.3: verify ssh service

```

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

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

khlvn@workstation:~$ █

```

```

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

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

```

```

khlvn@ubuntuKHLVN:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
khlvn@ubuntuKHLVN:~$ sudo ufw enable
sudo uFirewall is active and enabled on system startup
khlvn@ubuntuKHLVN:~$ sudo ufw status
Status: active

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

```

figure 2.4: configure the firewall to all port 22

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: 192.168.56.10
 - 1.2 Server 2 IP address: 192.168.56.11
 - 1.3 Server 3 IP address: 192.168.56.12

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**

prior to this task, I have created an internal network in VirtualBox for the three VMs to use with the following command in command prompt:

```
> vboxmanage dhcpserver add --network=sysadnetwork --server-ip=192.168.56.1  
--lower-ip=192.168.56.10 --upper-ip=192.168.56.60 --netmask=255.255.255.0  
--enable
```

Task 3 Documentation

```
khln@workstation:~$ ifconfig  
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.56.10 netmask 255.255.255.0 broadcast 192.168.56.255  
    inet6 fe80::7c5d:9278:9a9c:2b5b prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:e8:2b:3a txqueuelen 1000 (Ethernet)  
    RX packets 1 bytes 590 (590.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 36 bytes 4427 (4.4 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 264 bytes 20094 (20.0 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 264 bytes 20094 (20.0 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```

khlvn@server1:~$ ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.11 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::e17d:5d0c:7693:4ed2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a1:46:bc txqueuelen 1000 (Ethernet)
    RX packets 40 bytes 7133 (7.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 46 bytes 5651 (5.6 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 1228 bytes 88474 (88.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1228 bytes 88474 (88.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

khlvn@server2:~$ ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.12 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::4bbf:dd53:ec0a:15cd prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:38:3b:94 txqueuelen 1000 (Ethernet)
    RX packets 23 bytes 4463 (4.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44 bytes 5343 (5.3 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 746 bytes 54280 (54.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 746 bytes 54280 (54.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

figure 3.1: recording the ip address of each VM

```

khlvn@workstation:~$ ping 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=1.79 ms
64 bytes from 192.168.56.11: icmp_seq=2 ttl=64 time=0.361 ms
64 bytes from 192.168.56.11: icmp_seq=3 ttl=64 time=0.787 ms
64 bytes from 192.168.56.11: icmp_seq=4 ttl=64 time=0.906 ms
64 bytes from 192.168.56.11: icmp_seq=5 ttl=64 time=3.47 ms
^C
--- 192.168.56.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4587ms
rtt min/avg/max/mdev = 0.361/1.462/3.468/1.105 ms
khlvn@workstation:~$ ping 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=1.41 ms
64 bytes from 192.168.56.12: icmp_seq=2 ttl=64 time=1.27 ms
64 bytes from 192.168.56.12: icmp_seq=3 ttl=64 time=1.21 ms
64 bytes from 192.168.56.12: icmp_seq=4 ttl=64 time=0.952 ms
64 bytes from 192.168.56.12: icmp_seq=5 ttl=64 time=0.553 ms
^C
--- 192.168.56.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4818ms
rtt min/avg/max/mdev = 0.553/1.078/1.411/0.301 ms

```

figure 3.2.1: testing connectivity of local machine to the 2 servers

```

khlvn@server1:~$ ping 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=4.00 ms
64 bytes from 192.168.56.12: icmp_seq=2 ttl=64 time=0.629 ms
64 bytes from 192.168.56.12: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.56.12: icmp_seq=4 ttl=64 time=0.883 ms
64 bytes from 192.168.56.12: icmp_seq=5 ttl=64 time=1.08 ms
64 bytes from 192.168.56.12: icmp_seq=6 ttl=64 time=1.60 ms
^C
--- 192.168.56.12 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 6691ms
rtt min/avg/max/mdev = 0.629/1.533/4.003/1.142 ms

```

figure 3.2.2: testing connectivity from server1 to server2

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`
2. Logout of Server 1 by issuing the command `control + D`.
3. Do the same for Server 2.
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.
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.

Task 4 Documentation

```
khlvn@workstation:~$ ssh khlvn@192.168.56.11
khlvn@192.168.56.11'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

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.

khlvn@server1:~$ whoami
khlvn
khlvn@server1:~$ cat /etc/hostname
server1
khlvn@server1:~$
```

figure 4.1: using ssh to connect to server1 using local machine

```
khlvn@server1:~$  
logout  
Connection to 192.168.56.11 closed.
```

figure 4.2: closing the ssh session using the key Ctrl + D

```
khlvn@workstation:~$ ssh khlvn@192.168.56.12  
The authenticity of host '192.168.56.12 (192.168.56.12)' can't be established.  
ED25519 key fingerprint is SHA256:wnvwXh4tP0tdxbAt4RnonaiES1EMEWl0WMxhP76SZe4.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.56.12' (ED25519) to the list of known hosts.  
khlvn@192.168.56.12'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  
  
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.  
  
khlvn@server2:~$ whoami  
khlvn  
khlvn@server2:~$ cat /etc/hostname  
server2  
khlvn@server2:~$
```

figure 4.3: using ssh to connect to server2 using local machine

```
khlvn@workstation:~$ sudo nano /etc/hosts  
[sudo] password for khlvn:  
khlvn@workstation:~$ cat /etc/hosts  
# Standard host addresses  
127.0.0.1    workstation  
::1         localhost ip6-localhost ip6-loopback  
ff02::1     ip6-allnodes  
ff02::2     ip6-allrouters  
192.168.56.11 server1  
192.168.56.12 server2  
# This host address  
127.0.1.1   ubuntuKHLVN  
khlvn@workstation:~$
```

figure 4.4: editing the hosts of the local machine in /etc/hosts

```

khlvn@workstation:~$ ssh khlvn@server1
The authenticity of host 'server1 (192.168.56.11)' can't be established.
ED25519 key fingerprint is SHA256:tBAYQDD3r0iQXAWloz+/4coQmRgCzqGdHQ93GevWP3M.
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.
khlvn@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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 18:00:16 2024 from 192.168.56.10
khlvn@server1:~$ uname -a
Linux server1 6.8.0-41-generic #41-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 2 20:41:06 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
khlvn@server1:~$

```

figure 4.5.1: testing ssh connectivity of server1 using the hostname

```

khlvn@workstation:~$ ssh khlvn@server2
The authenticity of host 'server2 (192.168.56.12)' can't be established.
ED25519 key fingerprint is SHA256:wnvwXh4tP0tdxbAt4RnonaIES1EMEWL0WmxhP76SZe4.
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.
khlvn@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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 18:03:54 2024 from 192.168.56.10
khlvn@server2:~$

```

figure 4.5.2: testing ssh connectivity of server2 using the hostname

Reflections:

Answer the following:

- How are we able to use the hostname instead of IP address in SSH commands?
 - By following the criteria that the server is in the same network and the hostname of the servers are set in the `/etc/hosts` file of the local machine, then we can use the hostname of the server to connect remotely to the servers using SSH instead of using the given IP Address.
- How secure is SSH?
 - SSH uses a strong hashing algorithm and encryption and it is a better alternative remote shell protocol than telnet, which has no security. When intercepting a telnet packet using packet analyzing tools like Wireshark, we can see the plain password of a remote user, comparing it to ssh packet, it will display the contents of the packet as random characters.