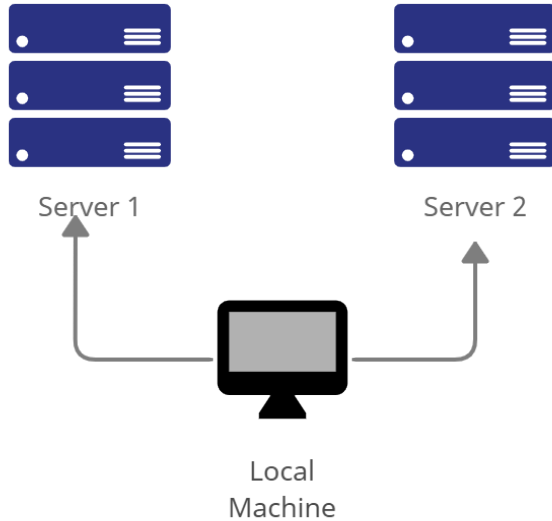


Name: Karlo D. Santos	Date Performed: 08/22/2023
Course/Section: CPE31S5	Date Submitted: 08/23/2023
Instructor: Engr. Roman Richard	Semester and SY: 1st (2023-2024)
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 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, <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>).	
	
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	
<pre>santos@kdlsantos:~\$ sudo nano /etc/hostname</pre>	

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

GNU nano 6.2 /etc/hostname *
Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo

santos@server1:~$
```

1.2 Use server2 for Server 2

```
santos2@kdlsantos2:~$ sudo nano /etc/hostname
[sudo] password for santos2:
```

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

GNU nano 6.2 /etc/hostname *
Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo

santos2@server2:~$ _
```

1.3 Use workstation for the Local Machine

```
santos@kdisantos-VirtualBox:~$ sudo nano /etc/hostname
[sudo] password for santos:

GNU nano 6.2 /etc/hostname *
workstation

^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

santos@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

```
santos@server1:~$ sudo nano /etc/hosts

GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 kdisantos

# 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

[ Read 9 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_/ Go To Line M-E Redo
```

2.2 Type 127.0.0.1 server 2 for Server 2

```
santos2@server2:~$ sudo nano /etc/hosts
[sudo] password for santos2: _
```

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 kdisantos2

# 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

[ Read 9 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line M-E
```

2.3 Type 127.0.0.1 workstation for the Local Machine

```
santos@workstation:~$ sudo nano /etc/hosts
[sudo] password for santos:
```

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 kdisantos-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

[ Read 9 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_ Go To Line
```

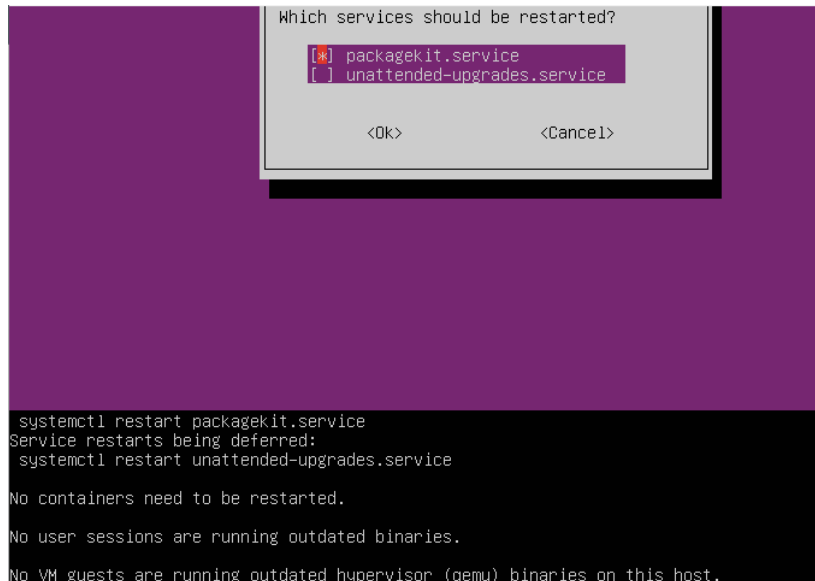
Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

1. Upgrade the packages by issuing the command ***sudo apt update*** and ***sudo apt upgrade*** respectively.

Server 1

```
santos@server1:~$ 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.
santos@server1:~$
```

```
santos@server1:~$ 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.
```



Which services should be restarted?

- ☒ packagekit.service
- ☐ unattended-upgrades.service

<Ok> <Cancel>

```
systemctl restart packagekit.service
Service restarts being deferred:
systemctl restart unattended-upgrades.service

No containers need to be restarted.

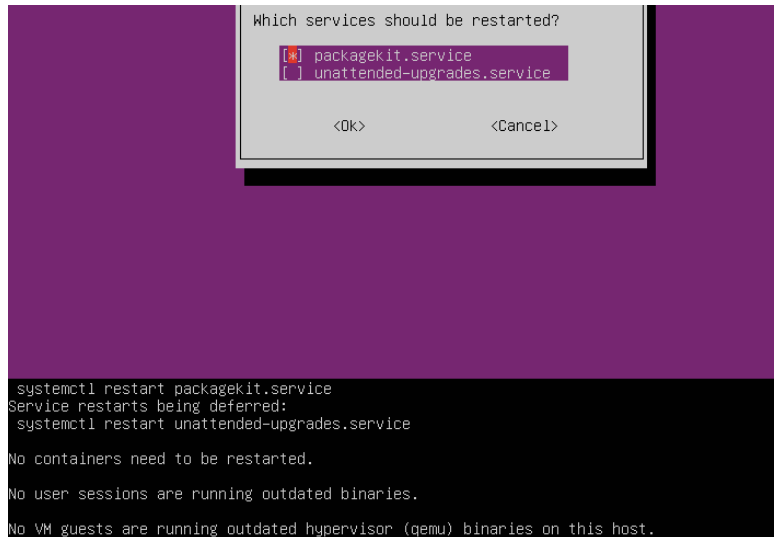
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Server 2

```
santos2@server2:~$ 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.
```

```
santos2@server2:~$ 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 7747 kB of archives.
After this operation, 838 kB disk space will be freed.
```



```
systemctl restart packagekit.service
Service restarts being deferred:
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Local machine

```
santos@workstation:~$ 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://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
26 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
santos@workstation:~$ 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-lsmb vim-common vim-tiny
  xxd
```

```
Setting up ghostscript-x (9.55.0~dfsg1-0ubuntu5.4) ...
Processing triggers for mailcap (3.70~nmulubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for initramfs-tools (0.140ubuntu13.4) ...
update-initramfs: Generating /boot/initrd.img-6.2.0-26-generic
```

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

Server 1

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

Server 2

```
santos2@server2:~$ 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.
santos2@server2:~$
```

Local machine

```
santos@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.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.3 [38.8 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.3 [434 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]

```

3. Verify if the SSH service has started by issuing the following commands:

3.1 *sudo service ssh start*

Server 1

```
santos@server1:~$ sudo service ssh start
santos@server1:~$ _
```

Server 2

```
santos2@server2:~$ sudo service ssh start
santos2@server2:~$ _
```

Local machine

```
santos@workstation:~$ sudo service ssh start
santos@workstation:~$
```

3.2 *sudo systemctl status ssh*

Server 1

```
santos@server1:~$ 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 15:46:58 UTC; 2min 49s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 674 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 717 (sshd)
    Tasks: 1 (limit: 4557)
   Memory: 4.4M
      CPU: 28ms
   CGroup: /system.slice/ssh.service
           └─717 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 15:46:57 server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 15:46:58 server1 sshd[717]: Server listening on 0.0.0.0 port 22.
Aug 22 15:46:58 server1 sshd[717]: Server listening on :: port 22.
Aug 22 15:46:58 server1 systemd[1]: Started OpenBSD Secure Shell server.
santos@server1:~$
```

Server 2

```
santos2@server2:~$ 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 16:19:52 UTC; 1min 38s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 1118 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1119 (sshd)
    Tasks: 1 (limit: 4557)
   Memory: 1.7M
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─1119 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 16:19:52 server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 16:19:52 server2 sshd[1119]: Server listening on 0.0.0.0 port 22.
Aug 22 16:19:52 server2 sshd[1119]: Server listening on :: port 22.
Aug 22 16:19:52 server2 systemd[1]: Started OpenBSD Secure Shell server.
santos2@server2:~$
```

Local machine

```
santos@workstation:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e
   Active: active (running) since Wed 2023-08-23 00:27:29 PST; 1min 5s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 3548 (sshd)
    Tasks: 1 (limit: 4591)
   Memory: 1.7M
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─3548 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 23 00:27:29 workstation systemd[1]: Starting OpenBSD Secure Shell server...
Aug 23 00:27:29 workstation sshd[3548]: Server listening on 0.0.0.0 port 22.
Aug 23 00:27:29 workstation sshd[3548]: Server listening on :: port 22.
Aug 23 00:27:29 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*

Server 1

```
santos@server1:~$ sudo ufw allow ssh
[sudo] password for santos:
Rules updated
Rules updated (v6)
```


Server 2

```
santos2@server2:~$ sudo ufw allow ssh
[sudo] password for santos2:
Rules updated
Rules updated (v6)
```

Local machine

```
santos@workstation:~$ sudo ufw allow ssh
[sudo] password for santos:
Rules updated
Rules updated (v6)
```

4.2 *sudo ufw enable*

Server 1

```
santos@server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Server 2

```
santos2@server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
santos2@server2:~$ sudo ufw status
```

Local machine

```
santos@workstation:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

4.3 *sudo ufw status*

Server 1

```
santos@server1:~$ sudo ufw status
Status: active

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

Server 2

```
santos2@server2:~$ sudo ufw status
Status: active

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

Local machine

```
santos@workstation:~$ 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: 192.168.56.101

```
santos@server1:~$ 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::a00:27ff:feff:5226 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ff:52:26 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 3038 (3.0 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 564 bytes 40300 (40.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 564 bytes 40300 (40.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

1.2 Server 2 IP address: 192.168.56.104

```
santos2@server2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe55:f800 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:55:f8:00 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 3038 (3.0 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 404 bytes 28940 (28.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 404 bytes 28940 (28.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

1.3 Server 3 IP address: 192.168.56.103

```
santos@workstation:~$ 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::87e7:7f49:4b1a:8d41 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:be:a7:c1 txqueuelen 1000 (Ethernet)
    RX packets 3 bytes 1770 (1.7 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 883 bytes 65255 (65.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 883 bytes 65255 (65.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. Make sure that they can ping each other.

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

```
santos@workstation:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=1.41 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.853 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=1.23 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=1.36 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=1.39 ms
^C
--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 0.853/1.247/1.412/0.207 ms
```

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

```
santos@workstation:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.985 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=1.12 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.456 ms
64 bytes from 192.168.56.104: icmp_seq=5 ttl=64 time=1.02 ms
^C
--- 192.168.56.104 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4035ms
rtt min/avg/max/mdev = 0.456/1.047/1.657/0.382 ms
```

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

```
santos@server1:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=1.32 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.790 ms
64 bytes from 192.168.56.104: icmp_seq=5 ttl=64 time=0.969 ms
64 bytes from 192.168.56.104: icmp_seq=6 ttl=64 time=0.866 ms
^C
--- 192.168.56.104 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.790/1.086/1.354/0.221 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`

```
santos@workstation:~$ ssh santos@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:gp0Utio2p0xeEN/5c3xV2GEpprN2oCOEB0hFy2dnG3w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

1.2 Enter the password for server 1 when prompted

```
santos@workstation:~$ ssh santos@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:gp0Utio2p0xeEN/5c3xV2GEpprN2oCOEB0hFy2dnG3w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
santos@192.168.56.101's password:
```

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

```
santos@server1:~$
```

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

```
santos@server1:~$
logout
Connection to 192.168.56.101 closed.
```

3. Do the same for Server 2.

```
santos@workstation:~$ ssh santos@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:L6ra3iMwops4PlffcZuokKT/DInrb+/ibNy+SVRF3EA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known hosts.
santos@192.168.56.104's password:
```

```

santos2@192.168.56.104'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 17:48:21 UTC 2023

System load:  0.009765625      Processes:            123
Usage of /:   39.5% of 15.64GB  Users logged in:     1
Memory usage: 5%              IPv4 address for enp0s3: 192.168.56.104
Swap usage:   0%

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 22 17:44:57 2023
santos2@server2:~$
logout
Connection to 192.168.56.104 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)

4.2 *IP_address server 2* (provide the ip address of server 2 followed by the hostname)

```

GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 kdlsantos-VirtualBox

192.168.56.101 server1
192.168.56.104 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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Locati
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify  ^_ Go To

```

4.3 Save the file and exit.

```

[Wrote 11 lines]
^W Where Is  ^K Cut       ^T Execute
^_ Replace   ^U Paste     ^J Justify

```

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.

Server 1

```
santos@workstation:~$ ssh santos@server1
The authenticity of host 'server1 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:gp0Utio2p0XeEN/5c3xV2GEpprN2oCOEB0hFy2dnG3w.
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.
santos@server1'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 05:54:24 PM UTC 2023

System load:  0.1064453125      Processes:           112
Usage of /:   40.7% of 14.30GB   Users logged in:     1
Memory usage: 6%               IPv4 address for enp0s3: 192.168.56.101
Swap usage:   0%

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 22 17:39:29 2023 from 192.168.56.103
santos@server1:~$
logout
Connection to server1 closed.
```

Server 2

```
santos@workstation:~$ ssh santos@server2
The authenticity of host 'server2 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:L6ra3iMwops4PlffcZuokKT/DInrb+/ibNy+SVRF3EA.
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
```

```
santos2@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 17:55:29 UTC 2023

System load:  0.0732421875      Processes:           116
Usage of /:   39.5% of 15.64GB   Users logged in:     1
Memory usage: 5%                IPv4 address for enp0s3: 192.168.56.104
Swap usage:   0%

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 22 17:48:21 2023 from 192.168.56.103
santos2@server2:~$
logout
Connection to server2 closed.
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
 - **Based on the activity, we are able to use hostname instead of IP address in SSH commands by making sure it was included or declared in the /etc/hosts. One example of this is the number 4 in task 4 where we use `sudo nano /etc/hosts`, and add the text that includes the IP address and the server number that will serve as hostname. Since the SSH client depends on the Domain Name System(DNS) in converting the hostname into an IP address. So it's important to make sure that the hostname will undergo in DNS by including it in /etc/hosts or make sure that the hostname is configured in the DNS network server.**
2. How secured is SSH?
 - **The Secure Shell (SSH) is a secured method in terms of remote access and other things like data communication. Since it has an SSH protocol that uses a method of encryption in making sure of the security in the connection between the client and the server. Another one, is the authentication that can be found in using this, like password and public-keys authentication. It includes the host keys that SSH used too verify the authenticity of the server. Like what we do in the activity where**

**we enter the password of the server when we connect it for the first time.
It helps in making sure of the security of the server.**