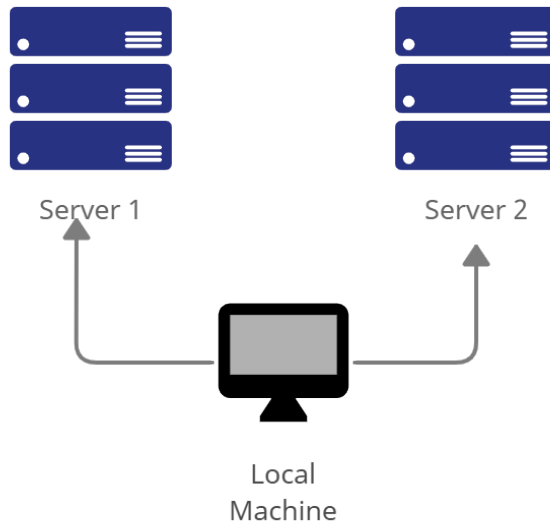


Name:Calderon Ricardo B.	Date Performed:01-16-24
Course/Section:CPE31S1	Date Submitted:01-23-24
Instructor: Dr. Jonathan Taylar	Semester and SY: 2nd sem 2023-2024
Activity 1: Configure Network using Virtual Machines	
1. Objectives: 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 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>).	
 <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 stacks. The left server stack is labeled "Server 1" and the right server stack is labeled "Server 2". Each server stack consists of three blue rectangular blocks, each with a white dot and three horizontal lines, representing a server rack.</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	

```
GNU nano 6.2 /etc/hostname
Server 1

[ Read 2 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

calderon@calderon:~$ sudo nano /etc/hostname
[sudo] password for calderon:
calderon@calderon:~$ reboot

calderon@Server1: ~
calderon@Server1:~$
calderon@Server1:~$
calderon@Server1:~$
```

1.2 Use server2 for Server 2

```
calderon@calderon: ~
GNU nano 6.2 /etc/hostname *
Server 2

File Name to Write: /etc/hostname
^G Help      M-D DOS Format M-A Append   M-B Backup File
^C Cancel    M-M Mac Format M-P Prepend  ^T Browse

calderon@calderon:~$ sudo nano /etc/hostname
[sudo] password for calderon:
calderon@calderon:~$ reboot

calderon@Server2: ~
calderon@Server2:~$
calderon@Server2:~$
calderon@Server2:~$
```

1.3 Use workstation for the Local Machine

```
calderon@calderon: ~
calderon@calderon:~$ sudo nano /etc/hostname
```

```
calderon@calderon: ~  
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  
  
calderon@calderon:~$ reboot  
  
calderon@workstation: ~  
calderon@workstation:~$  
calderon@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

```
calderon@Server1:~$ sudo nano /etc/hosts  
[sudo] password for calderon:  
calderon@Server1:~$
```

```
calderon@Server1: ~  
GNU nano 6.2 /etc/hosts *  
127.0.0.1 localhost  
127.0.0.1 Server 1  
  
# 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  
  
File Name to Write: /etc/hosts  
^G Help      M-D DOS Format  M-A Append     M-B Backup File  
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

- 2.2 Type 127.0.0.1 server 2 for Server 2

```
calderon@Server2: ~  
calderon@Server2:~$ sudo nano /etc/hosts  
[sudo] password for calderon:
```

```
calderon@Server2: ~  
GNU nano 6.2 /etc/hosts *  
127.0.0.1 localhost  
127.0.0.1 Server 2  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
File Name to Write: /etc/hosts  
^G Help M-D DOS Format M-A Append M-B Backup File  
^C Cancel M-M Mac Format M-P Prepend ^T Browse
```

2.3 Type 127.0.0.1 workstation for the Local Machine

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

```
calderon@workstation: ~  
GNU nano 6.2 /etc/hosts *  
\127.0.0.1 localhost  
127.0.0.1 workstation  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
File Name to Write: /etc/hosts  
^G Help M-D DOS Format M-A Append M-B Backup File  
^C Cancel M-M Mac Format M-P Prepend ^T Browse
```

2.4

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,

```
calderon@Server1: ~  
calderon@Server1:~$ sudo apt update | sudo apt upgrade -y  
[sudo] password for calderon:  
  
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  
The following NEW packages will be installed:  
  ubuntu-pro-client-l10n  
The following packages have been kept back:  
  gjs libgjs0g tzdata  
The following packages will be upgraded:  
  alsa-ucm-conf amd64-microcode apparmor apt apt-utils avahi-autoipd  
  avahi-daemon avahi-utils bind9-dnsutils bind9-host bind9-libs bluez  
  bluez-cups bluez-obexd cups cups-bsd cups-client cups-common  
  cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common  
  distro-info distro-info-data dnsmasq-base evince evince-common file  
  firmware-sof-signed fonts-noto-color-emoji fonts-opensymbol ghostscript  
  ghostscript-x qir1.2-javascriptcoregtk-4.0 qir1.2-mutter-10
```

Server 2,

```
calderon@Server2: ~  
calderon@Server2:~$ sudo apt update | sudo apt upgrade -y  
[sudo] password for calderon:  
  
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.  
  
E: Invalid operation update  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following NEW packages will be installed:  
  ubuntu-pro-client-l10n  
The following packages have been kept back:  
  gjs libgjs0g tzdata  
The following packages will be upgraded:  
  alsa-ucm-conf amd64-microcode apparmor apt apt-utils avahi-autoipd  
  avahi-daemon avahi-utils bind9-dnsutils bind9-host bind9-libs bluez  
  bluez-cups bluez-obexd cups cups-bsd cups-client cups-common  
  cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common  
  distro-info distro-info-data dnsmasq-base evince evince-common file  
  firmware-sof-signed fonts-noto-color-emoji fonts-opensymbol ghostscript
```

Local Machine,

```
calderon@workstation: ~  
calderon@workstation:~$ sudo apt update | sudo apt upgrade -y  
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  
The following NEW packages will be installed:  
  ubuntu-pro-client-l10n  
The following packages have been kept back:  
  gjs libgjs0g tzdata  
The following packages will be upgraded:  
  alsa-ucm-conf amd64-microcode apparmor apt apt-utils avahi-autoipd  
  avahi-daemon avahi-utils bind9-dnsutils bind9-host bind9-libs bluez  
  bluez-cups bluez-obexd cups cups-bsd cups-client cups-common  
  cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common  
  distro-info distro-info-data dnsmasq-base evince evince-common file  
  firmware-sof-signed fonts-noto-color-emoji fonts-opensymbol ghostscript  
  ghostscript-x gir1.2-javascriptcoregtk-4.0 gir1.2-mutter-10  
  gir1.2-webkit2-4.0 gnome-control-center gnome-control-center-data  
  gnome-control-center-faces gnome-remote-desktop initramfs-tools
```

2. Install the SSH server using the command *sudo apt install openssh-server*.
Server 1,

```
calderon@Server1: ~  
calderon@Server1:~$ 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 17 not upgraded.  
Need to get 752 kB of archives.  
After this operation, 6,050 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-  
server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-serve  
r amd64 1:8.9p1-3ubuntu0.6 [435 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term  
all 6.3-2ubuntu0.1 [267 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.1
```

Server 2,

```
calderon@Server2: ~  
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 3 not upgraded.  
Need to get 752 kB of archives.  
After this operation, 6,050 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-  
server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-serve  
r amd64 1:8.9p1-3ubuntu0.6 [435 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term  
all 6.3-2ubuntu0.1 [267 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.1  
1-0ubuntu1 [10.1 kB]  
Fetched 752 kB in 4s (180 kB/s)  
Preconfiguring packages ...  
Selecting previously unselected package openssh-sftp-server.  
[Reading database ... 80%
```

Local Machine,

```
calderon@workstation: ~  
calderon@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 17 not upgraded.  
Need to get 752 kB of archives.  
After this operation, 6,050 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-  
server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-serve  
r amd64 1:8.9p1-3ubuntu0.6 [435 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term  
all 6.3-2ubuntu0.1 [267 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.1  
1-0ubuntu1 [10.1 kB]  
Fetched 752 kB in 4s (180 kB/s)
```


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

3.1 *sudo service ssh start*

Server 1,

```
calderon@Server1: ~  
calderon@Server1:~$ sudo service ssh start  
calderon@Server1:~$
```

Server 2,

```
calderon@Server2: ~  
calderon@Server2:~$ sudo service ssh start  
calderon@Server2:~$
```

Local Machine.

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

3.2 *sudo systemctl status ssh*

Server 1,

```
calderon@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 2024-01-23 10:21:26 +08; 11min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 27751 (sshd)  
     Tasks: 1 (limit: 2260)  
    Memory: 1.7M  
       CPU: 60ms  
    CGroup: /system.slice/ssh.service  
            └─27751 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Jan 23 10:21:26 Server1 systemd[1]: Starting OpenBSD Secure Shell server...  
Jan 23 10:21:26 Server1 sshd[27751]: Server listening on 0.0.0.0 port 22.  
Jan 23 10:21:26 Server1 sshd[27751]: Server listening on :: port 22.  
Jan 23 10:21:26 Server1 systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-16/16 (END)
```

Server 2,


```
calderon@Server2:~$ 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 Tue 2024-01-23 10:23:29 +08; 17min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 27162 (sshd)
      Tasks: 1 (limit: 2260)
     Memory: 1.7M
        CPU: 40ms
    CGroup: /system.slice/ssh.service
            └─27162 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 23 10:23:29 Server2 systemd[1]: Starting OpenBSD Secure Shell server...
Jan 23 10:23:29 Server2 sshd[27162]: Server listening on 0.0.0.0 port 22.
Jan 23 10:23:29 Server2 sshd[27162]: Server listening on :: port 22.
Jan 23 10:23:29 Server2 systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

Local Machine.

```
calderon@workstation: ~
calderon@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 Tue 2024-01-23 10:18:06 +08; 21min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 27740 (sshd)
      Tasks: 1 (limit: 2260)
     Memory: 1.7M
        CPU: 47ms
    CGroup: /system.slice/ssh.service
            └─27740 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 23 10:18:06 workstation systemd[1]: Starting OpenBSD Secure Shell server...
Jan 23 10:18:06 workstation sshd[27740]: Server listening on 0.0.0.0 port 22.
Jan 23 10:18:06 workstation sshd[27740]: Server listening on :: port 22.
Jan 23 10:18:06 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,

```
calderon@Server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
calderon@Server1:~$
```

Server 2,

```
calderon@Server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
calderon@Server2:~$
```

Local Machine.

```
calderon@workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
calderon@workstation:~$
```

4.2 *sudo ufw enable*

Server 1,

```
calderon@Server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Server 2,

```
calderon@Server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Local Machine.

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

4.3 *sudo ufw status*

Server 1,

```
calderon@Server1:~$ sudo ufw status
Status: active

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

Server 2,

```
calderon@Server2:~$ sudo ufw status
Status: active

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

Local Machine.

```
calderon@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: 192.168.56.102

```
calderon@Server1:~$ ifconfig
enp0s3: 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::836e:6b60:5054:8e72 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:61:f7:23 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 176 bytes 15888 (15.8 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 411 bytes 34954 (34.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 411 bytes 34954 (34.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

1.2 Server 2 IP address: 192.168.56.103

```
calderon@Server2:~$ 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::836e:6b60:5054:8e72 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:61:f7:23 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 120 (120.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 107 bytes 10266 (10.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 358 bytes 29996 (29.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 358 bytes 29996 (29.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

calderon@Server2:~$
```

1.3 Server 3 IP address: 192.168.56.101

```
calderon@workstation:~$ 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::836e:6b60:5054:8e72 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:61:f7:23 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 360 (360.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 133 bytes 11997 (11.9 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 318 bytes 26988 (26.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 318 bytes 26988 (26.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

calderon@workstation:~$
```

2. Make sure that they can ping each other.

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

```
calderon@Server1:~$ 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=0.038 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.109 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.082 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.083 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.084 ms
64 bytes from 192.168.56.102: icmp_seq=7 ttl=64 time=0.083 ms
64 bytes from 192.168.56.102: icmp_seq=8 ttl=64 time=0.083 ms
64 bytes from 192.168.56.102: icmp_seq=9 ttl=64 time=0.087 ms
64 bytes from 192.168.56.102: icmp_seq=10 ttl=64 time=0.081 ms
64 bytes from 192.168.56.102: icmp_seq=11 ttl=64 time=0.081 ms
64 bytes from 192.168.56.102: icmp_seq=12 ttl=64 time=0.036 ms
64 bytes from 192.168.56.102: icmp_seq=13 ttl=64 time=0.083 ms
64 bytes from 192.168.56.102: icmp_seq=14 ttl=64 time=0.047 ms
64 bytes from 192.168.56.102: icmp_seq=15 ttl=64 time=0.032 ms
```

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

```
calderon@Server2:~$ 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=0.088 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.106 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.082 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.047 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.066 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.031 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=0.053 ms
64 bytes from 192.168.56.103: icmp_seq=8 ttl=64 time=0.080 ms
64 bytes from 192.168.56.103: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 192.168.56.103: icmp_seq=10 ttl=64 time=0.044 ms
```

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

```
calderon@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=0.418 ms
 64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.058 ms
 64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.052 ms
 64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.040 ms
 64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.056 ms
 64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.451 ms
 64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.132 ms
 64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.058 ms
 64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.074 ms
 64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.039 ms
 64 bytes from 192.168.56.101: icmp_seq=11 ttl=64 time=0.055 ms
 64 bytes from 192.168.56.101: icmp_seq=12 ttl=64 time=0.058 ms
 64 bytes from 192.168.56.101: icmp_seq=13 ttl=64 time=0.071 ms
 64 bytes from 192.168.56.101: icmp_seq=14 ttl=64 time=0.058 ms
 64 bytes from 192.168.56.101: icmp_seq=15 ttl=64 time=0.058 ms
 64 bytes from 192.168.56.101: icmp_seq=16 ttl=64 time=0.062 ms
 64 bytes from 192.168.56.101: icmp_seq=17 ttl=64 time=0.074 ms
 64 bytes from 192.168.56.101: icmp_seq=18 ttl=64 time=0.056 ms
 64 bytes from 192.168.56.101: icmp_seq=19 ttl=64 time=0.750 ms
 64 bytes from 192.168.56.101: icmp_seq=20 ttl=64 time=0.042 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

```
calderon@Server1:~$ ssh calderon@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:EsTZjx3wXkLwsyEReJv/M11gEtV55j3ISrkuffGpgP8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts
.
calderon@192.168.56.102's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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.

6 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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.
```


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

```
calderon@Server1:~$
```

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

```
calderon@Server1:~$  
logout  
Connection to 192.168.56.102 closed.  
calderon@Server1:~$
```

3. Do the same for Server 2.

```
calderon@Server2:~$ ssh calderon@192.168.56.103  
calderon@192.168.56.103's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
Last login: Tue Jan 23 13:52:10 2024 from 192.168.56.103  
calderon@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)


```
calderon@workstation: ~  
GNU nano 6.2 /etc/hosts *  
127.0.0.1 workstation  
192.168.56.102 server1  
192.168.56.103 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
```

4.3 Save the file and exit.

```
Save modified buffer?  
Y Yes  
N No ^C Cancel
```

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.

```
calderon@Server1:~$ ssh calderon@server1  
calderon@server1's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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.  
  
6 updates can be applied immediately.  
2 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
Last login: Tue Jan 23 14:10:19 2024 from 192.168.56.102  
calderon@Server1:~$
```

Do the same for Server 2.

```
calderon@Server2:~$ ssh calderon@server2
calderon@server2's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Tue Jan 23 14:14:21 2024 from 192.168.56.103
calderon@Server2:~$
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
We can use the hostname instead of IP address in SSH commands by editing the /etc/hosts` file to map the hostname to its IP address. This is a local file that provides a way of resolving hostnames to IP addresses without using DNS.

2. How secure is SSH?

SSH is widely used and considered secure due to its strong security features and resistance to attacks. However, it can still be vulnerable to certain types of attacks and best practices should be followed to maintain its security.

Conclusion

SSH is so important, because it is a secure protocol for remote access that provides strong security features such as encryption and authentication. It is widely used and resistant to various types of attacks. However, to maintain its security, best practices such as using strong passwords, disabling unused authentication methods, using firewall rules to restrict access, and keeping the SSH server software up-to-date with security patches should be followed.

