Name: Bartolome, Ken Christian Adrian V.	**Date Performed: 08/22/2023**
Course/Section: CPE31S5	**Date Submitted: 08/23/2023**
Instructor: Engr. Roman Richard	**Semester and SY: 1st Sem 2023-2024**

Activity 1: Configure Network using Virtual Machines

1. **Objectives:**

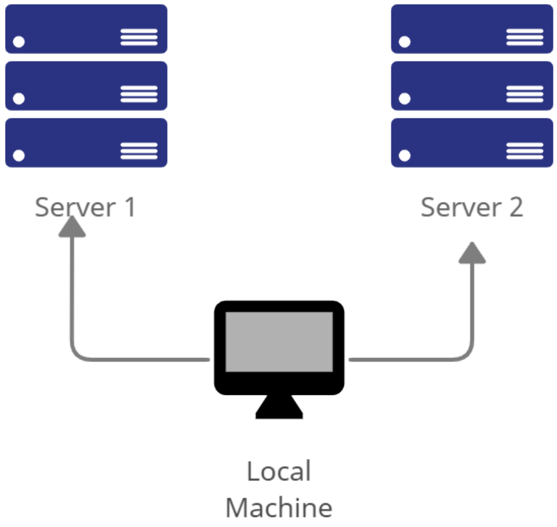
1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox

1.2. Set-up a Virtual Network and Test Connectivity of VMs

1. **Discussion:**

Network Topology:

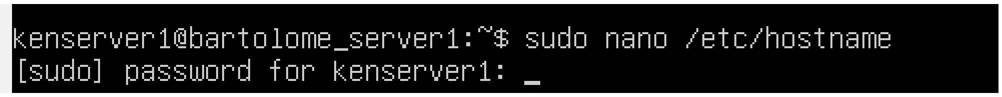
Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*).



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.

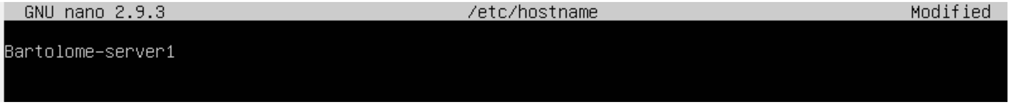
1. Change the hostname using the command *sudo nano /etc/hostname*

1.1 Use server1 for Server 1



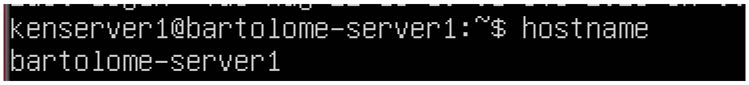
```
kenserver1@bartolome_server1:~$ sudo nano /etc/hostname
[sudo] password for kenserver1: _
```

1.1a sudo nano /etc/hostname server1



```
GNU nano 2.9.3                    /etc/hostname                    Modified

Bartolome-server1
```

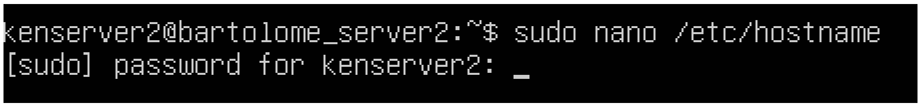
1.1b Saving



```
kenserver1@bartolome-server1:~$ hostname
bartolome-server1
```

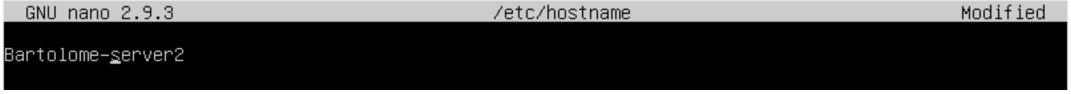
1.1c Checking

1.2 Use server2 for Server 2



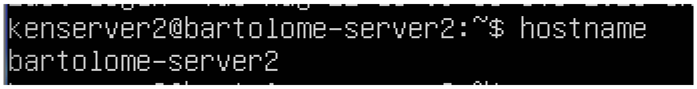
```
kenserver2@bartolome_server2:~$ sudo nano /etc/hostname
[sudo] password for kenserver2: _
```

1.2a sudo nano /etc/hostname server2



```
GNU nano 2.9.3                    /etc/hostname                    Modified

Bartolome-server2
```

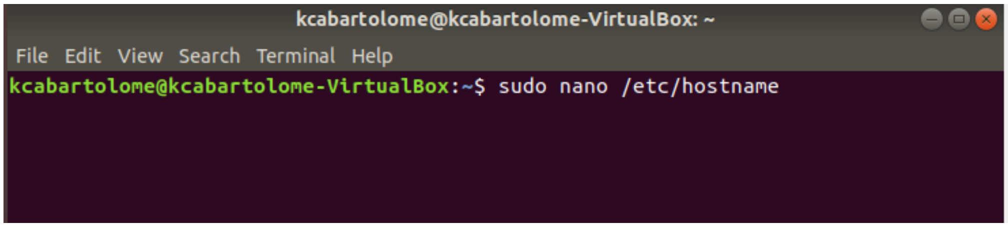
1.2b Rename



```
kenserver2@bartolome-server2:~$ hostname
bartolome-server2
```

1.2c Checking

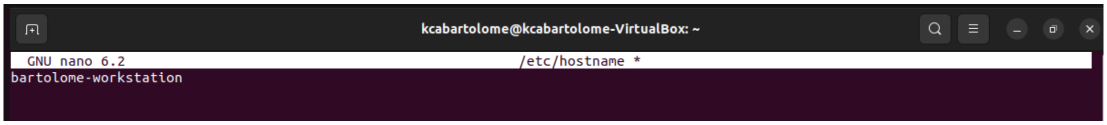
1.3 Use workstation for the Local Machine



kcabartolome@kcabartolome-VirtualBox: ~
File Edit View Search Terminal Help

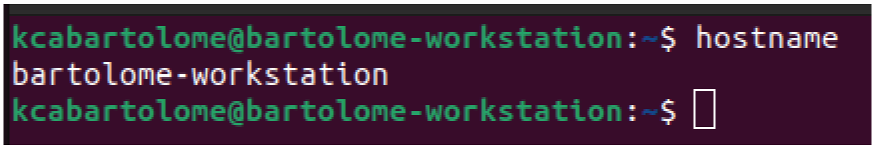
```
kcabartolome@kcabartolome-VirtualBox:~$ sudo nano /etc/hostname
```

1.3a sudo nano /etc/hostname workstation



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

1.3b Rename

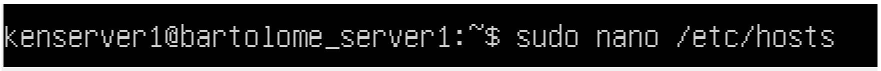


```
kcabartolome@bartolome-workstation:~$ hostname
bartolome-workstation
kcabartolome@bartolome-workstation:~$
```

1.3c Checking

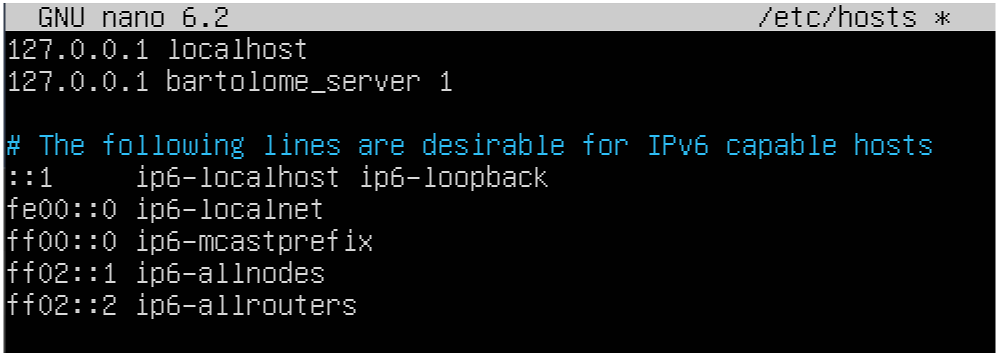
2. Edit the hosts using the command *sudo nano /etc/hosts*. Edit the second line.

2.1 Type 127.0.0.1 server 1 for Server 1



```
kenserver1@bartolome_server1:~$ sudo nano /etc/hosts
```

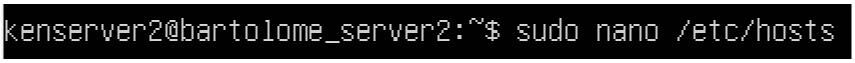
2.1a sudo nano /etc/hosts server1



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

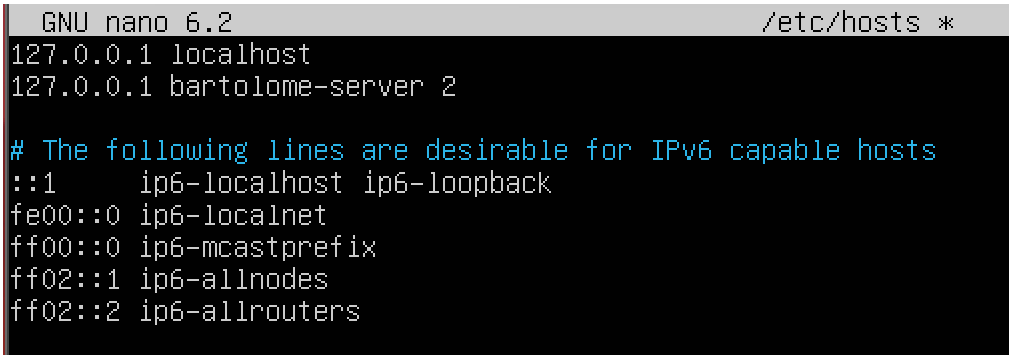
2.1b Rename

2.2 Type 127.0.0.1 server 2 for Server 2



```
kenserver2@bartolome_server2:~$ sudo nano /etc/hosts
```

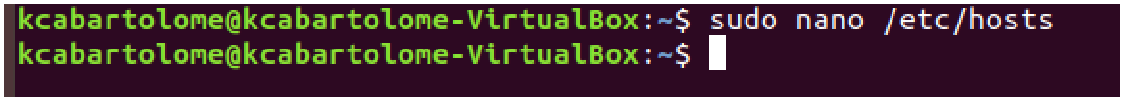
2.2a sudo nano /etc/hosts server2



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

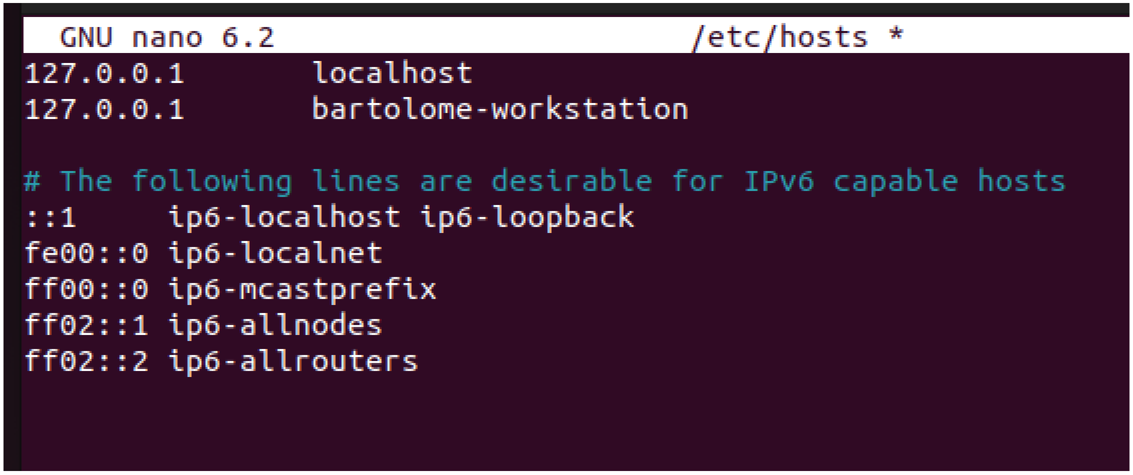
2.2b Rename

2.3 Type 127.0.0.1 workstation for the Local Machine



```
kcabartolome@kcabartolome-VirtualBox:~$ sudo nano /etc/hosts
kcabartolome@kcabartolome-VirtualBox:~$ 
```

2.3a sudo nano /etc/hosts workstation



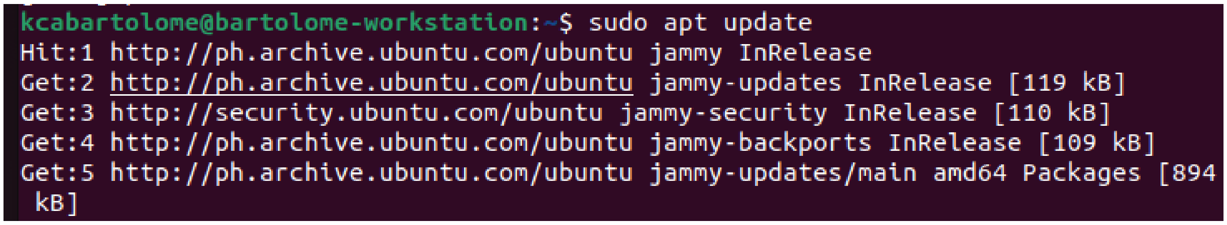
```
  GNU nano 6.2                         /etc/hosts *
127.0.0.1       localhost
127.0.0.1       bartolome-workstation

# 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
```

2.3b Rename

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

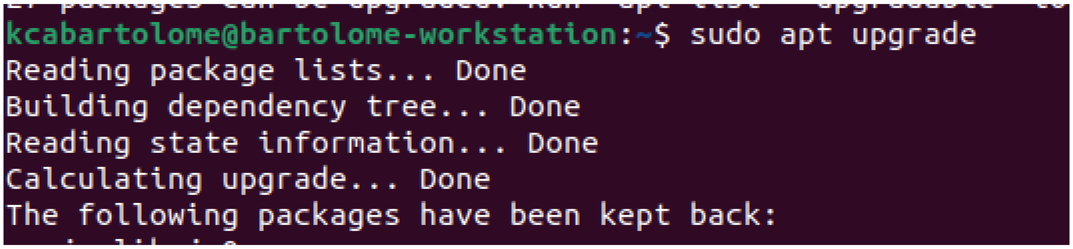
1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.

Workstation:



```
kcabartolome@bartolome-workstation:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [894
 kB]
```

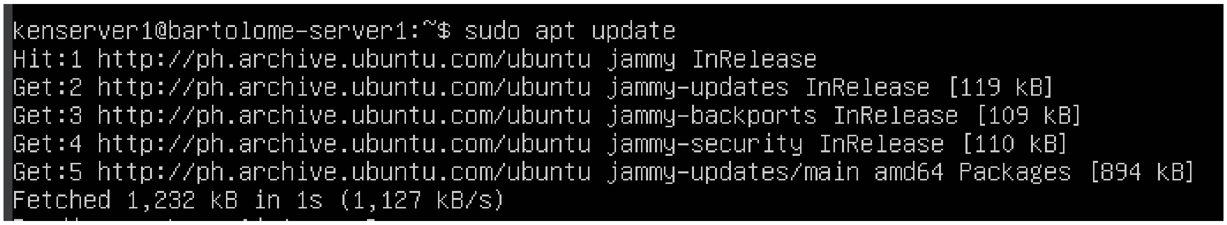
1.1a Workstation sudo apt update



```
kcabartolome@bartolome-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:
```

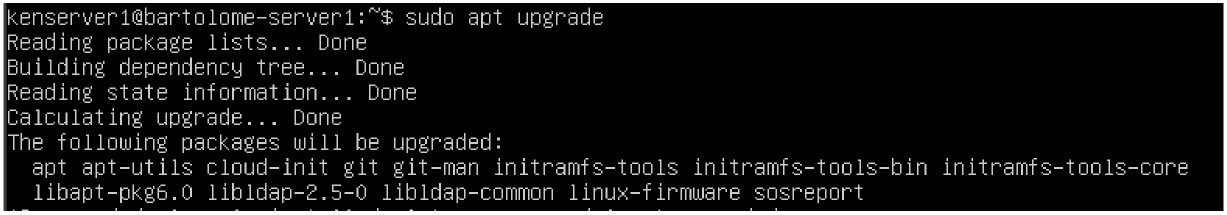
1.1b Workstation sudo apt upgrade

Server1:



```
kenserver1@bartolome-server1:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [894 kB]
Fetched 1,232 kB in 1s (1,127 kB/s)
```

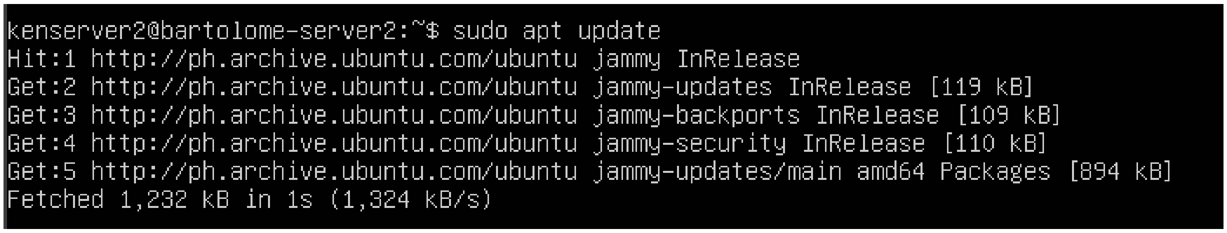
1.2a Server1 sudo apt update



```
kenserver1@bartolome-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 linux-firmware sosreport
```

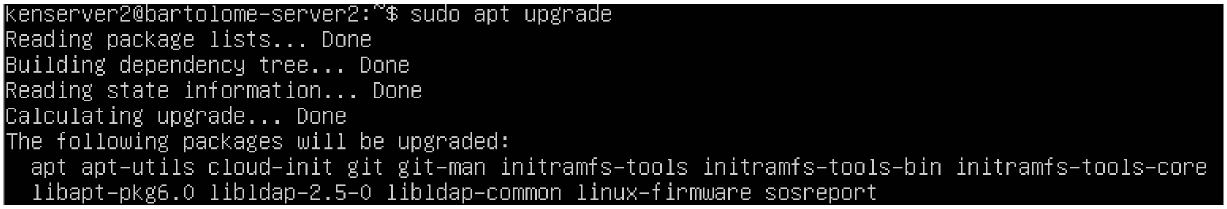
1.2b Server1 sudo apt upgrade

Server2:



```
kenserver2@bartolome-server2:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [894 kB]
Fetched 1,232 kB in 1s (1,324 kB/s)
```

1.3a Server2 sudo apt update

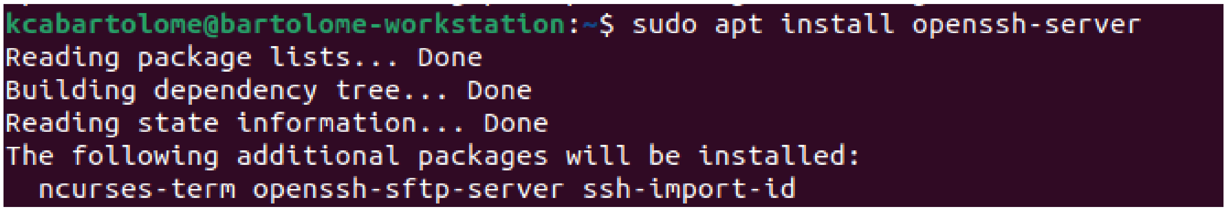


```
kenserver2@bartolome-server2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils cloud-init git git-man initramfs-tools initramfs-tools-bin initramfs-tools-core
  libapt-pkg6.0 libldap-2.5-0 libldap-common linux-firmware sosreport
```

1.3b Server2 sudo apt upgrade

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

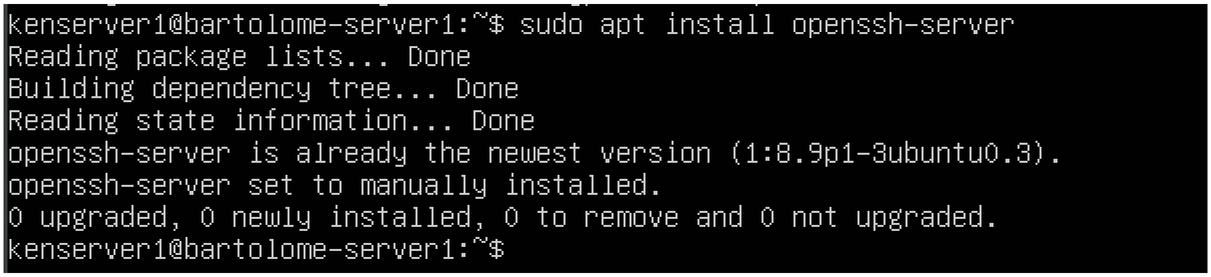
Workstation:



```
kcabartolome@bartolome-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
```

2.1a sudo apt install openssh-server (workstation)

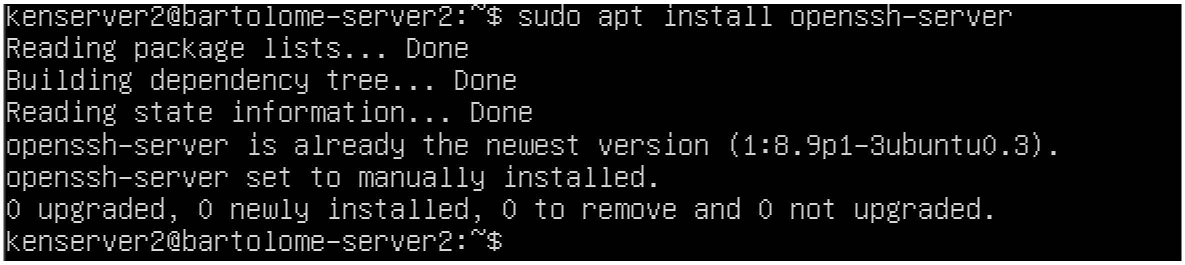
Server1:



```
kenserver1@bartolome-server1:~$ 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.
kenserver1@bartolome-server1:~$
```

2.1b sudo apt install openssh-server (server1)

Server2:



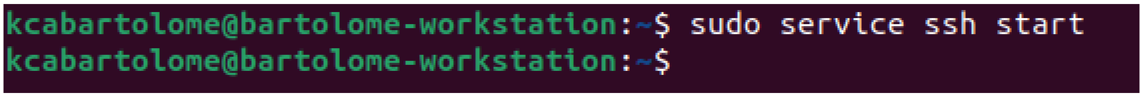
```
kenserver2@bartolome-server2:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.3).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
kenserver2@bartolome-server2:~$
```

2.1c sudo apt install openssh-server (server2)

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

   3.1 *sudo service ssh start*

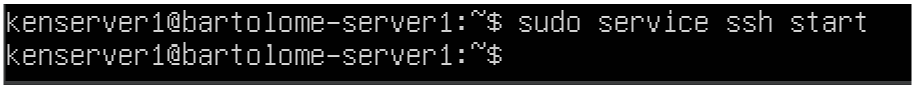
Workstation:



```
kcabartolome@bartolome-workstation:~$ sudo service ssh start
kcabartolome@bartolome-workstation:~$
```

3.1a sudo service ssh start (workstation)

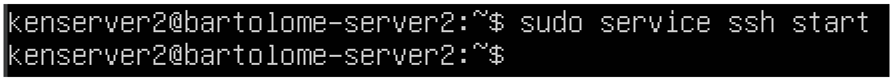
Server1:



```
kenserver1@bartolome-server1:~$ sudo service ssh start
kenserver1@bartolome-server1:~$
```

3.1b sudo service ssh start (server1)

Server2:

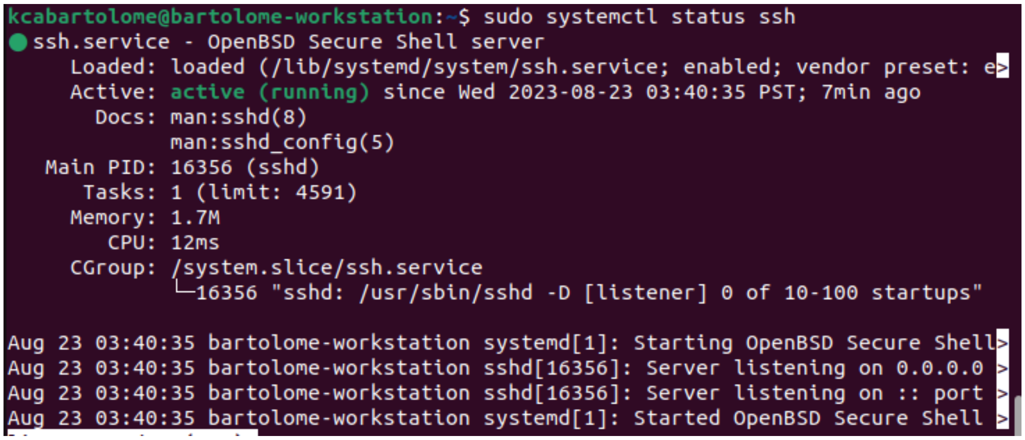


```
kenserver2@bartolome-server2:~$ sudo service ssh start
kenserver2@bartolome-server2:~$
```

3.1a sudo service ssh start (server2)

   3.2 *sudo systemctl status ssh*

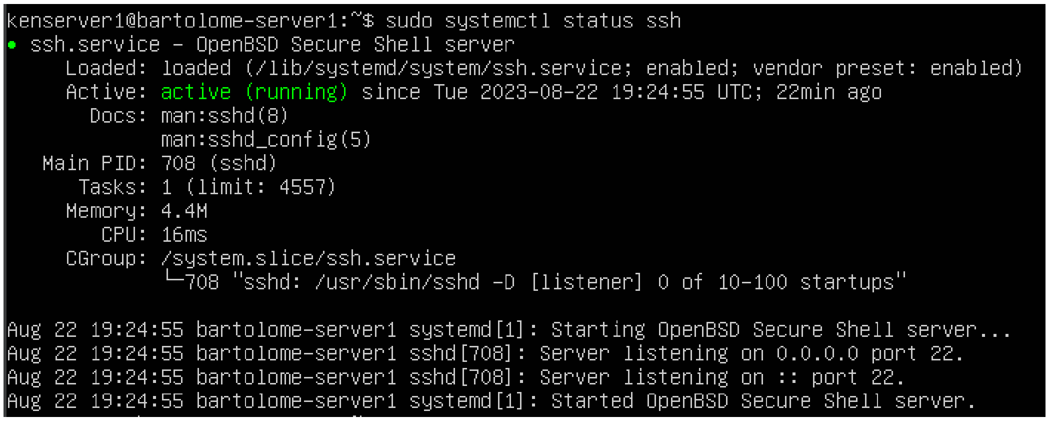
Workstation:



```
kcabartolome@bartolome-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 03:40:35 PST; 7min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 16356 (sshd)
      Tasks: 1 (limit: 4591)
     Memory: 1.7M
        CPU: 12ms
     CGroup: /system.slice/ssh.service
             └─16356 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 23 03:40:35 bartolome-workstation systemd[1]: Starting OpenBSD Secure Shell>
Aug 23 03:40:35 bartolome-workstation sshd[16356]: Server listening on 0.0.0.0 >
Aug 23 03:40:35 bartolome-workstation sshd[16356]: Server listening on :: port >
Aug 23 03:40:35 bartolome-workstation systemd[1]: Started OpenBSD Secure Shell >
```

3.2a sudo systemctl status ssh (workstation)

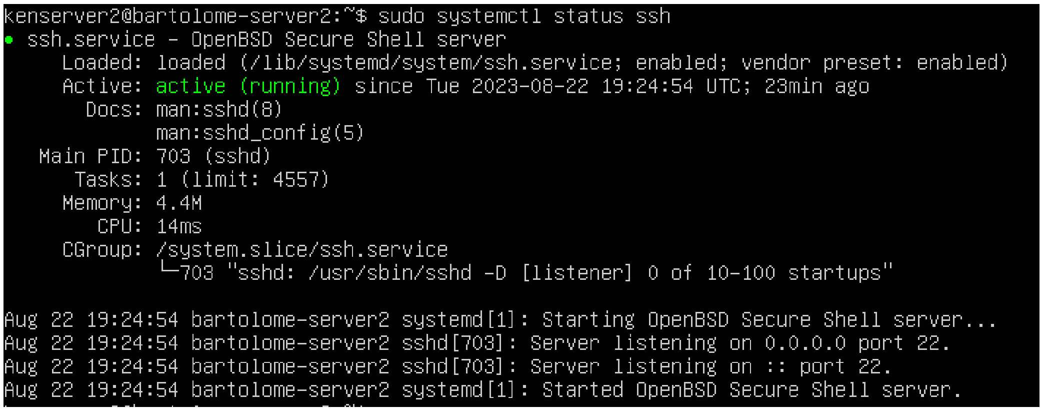
Server1:



```
kenserver1@bartolome-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 19:24:55 UTC; 22min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 708 (sshd)
      Tasks: 1 (limit: 4557)
     Memory: 4.4M
        CPU: 16ms
     CGroup: /system.slice/ssh.service
             └─708 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 19:24:55 bartolome-server1 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 19:24:55 bartolome-server1 sshd[708]: Server listening on 0.0.0.0 port 22.
Aug 22 19:24:55 bartolome-server1 sshd[708]: Server listening on :: port 22.
Aug 22 19:24:55 bartolome-server1 systemd[1]: Started OpenBSD Secure Shell server.
```

3.2b sudo systemctl status ssh (server1)

Server2:



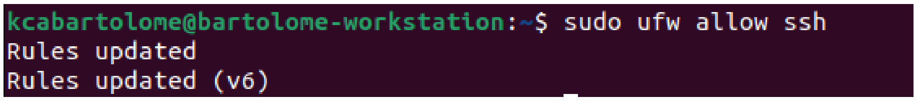
```
kenserver2@bartolome-server2:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-08-22 19:24:54 UTC; 23min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 703 (sshd)
      Tasks: 1 (limit: 4557)
     Memory: 4.4M
        CPU: 14ms
     CGroup: /system.slice/ssh.service
             └─703 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 22 19:24:54 bartolome-server2 systemd[1]: Starting OpenBSD Secure Shell server...
Aug 22 19:24:54 bartolome-server2 sshd[703]: Server listening on 0.0.0.0 port 22.
Aug 22 19:24:54 bartolome-server2 sshd[703]: Server listening on :: port 22.
Aug 22 19:24:54 bartolome-server2 systemd[1]: Started OpenBSD Secure Shell server.
```

3.2c sudo systemctl status ssh (server2)

4. Configure the firewall to all port 22 by issuing the following commands:

   4.1 *sudo ufw allow ssh*

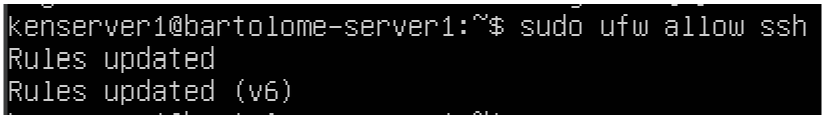
Workstation:



```
kcabartolome@bartolome-workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

4.1a sudo ufw allow ssh (workstation)

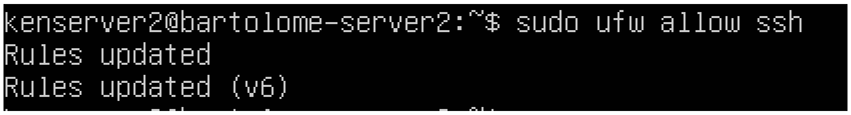
Server1:



```
kenserver1@bartolome-server1:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

4.1b sudo ufw allow ssh (server1)

Server2:

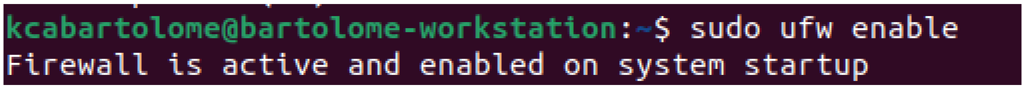


```
kenserver2@bartolome-server2:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

4.1c sudo ufw allow ssh (server2)

4.2 *sudo ufw enable*

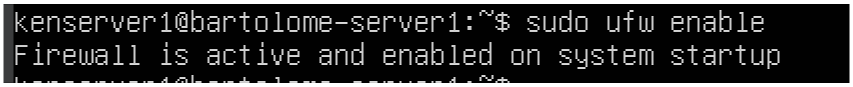
Workstation:



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

4.2a sudo ufw enable (workstation)

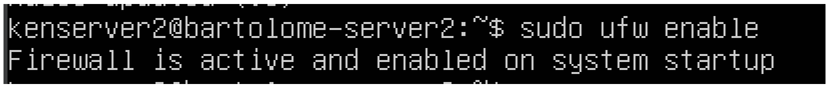
Server1:



```
kenserver1@bartolome-server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

4.2b sudo ufw enable (server1)

Server2:

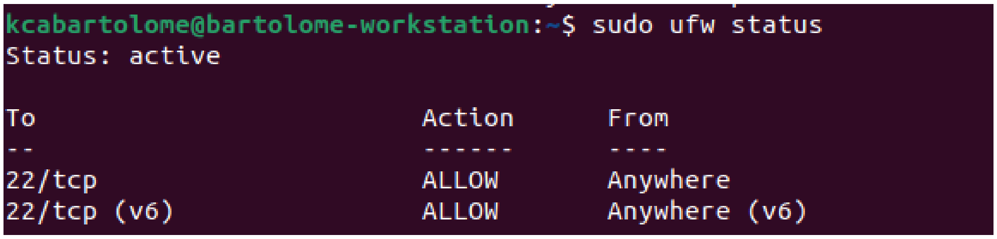


```
kenserver2@bartolome-server2:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

4.2c sudo ufw enable (server2)

4.3 *sudo ufw status*

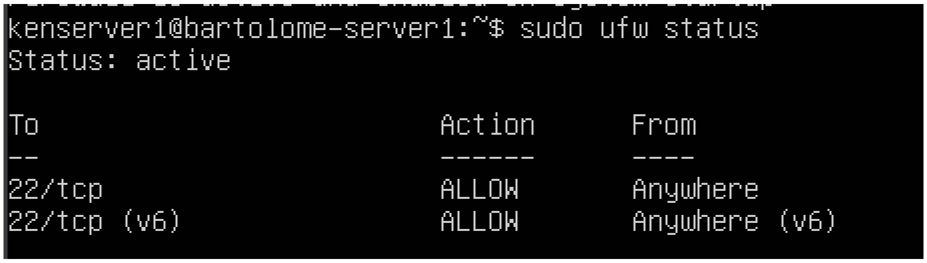
Workstation:



```
kcabartolome@bartolome-workstation:~$ sudo ufw status
Status: active

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

4.3a sudo ufw status (workstation)

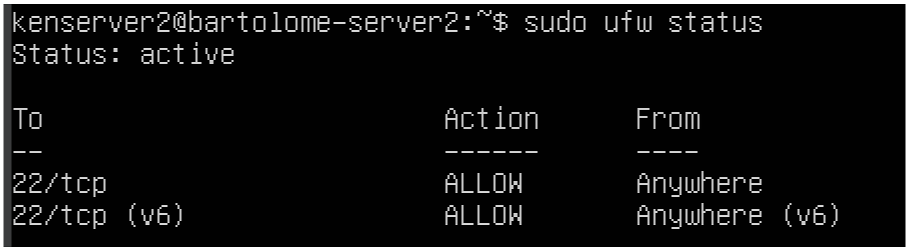
Server1:



```
kenserver1@bartolome-server1:~$ sudo ufw status
Status: active

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

4.3b sudo ufw status (server1)

Server2:



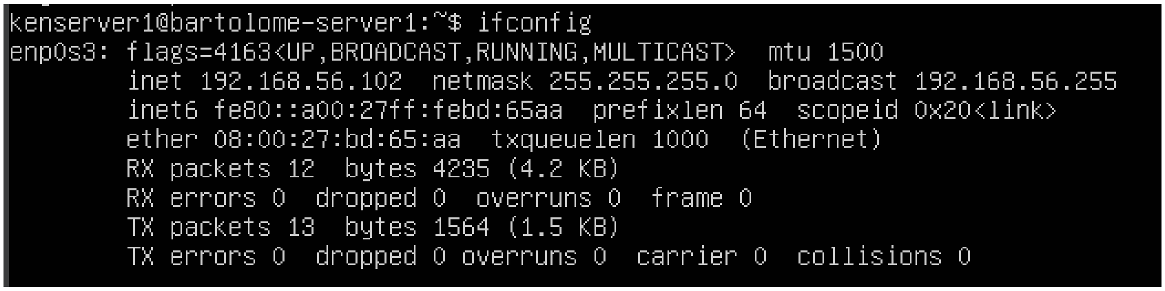
```
kenserver2@bartolome-server2:~$ sudo ufw status
Status: active

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

4.3c sudo ufw status (server2)

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

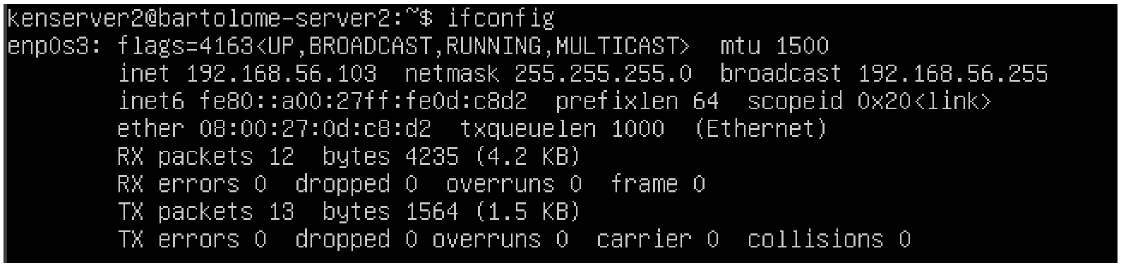
1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.

   1.1 Server 1 IP address: 192.168.56.102



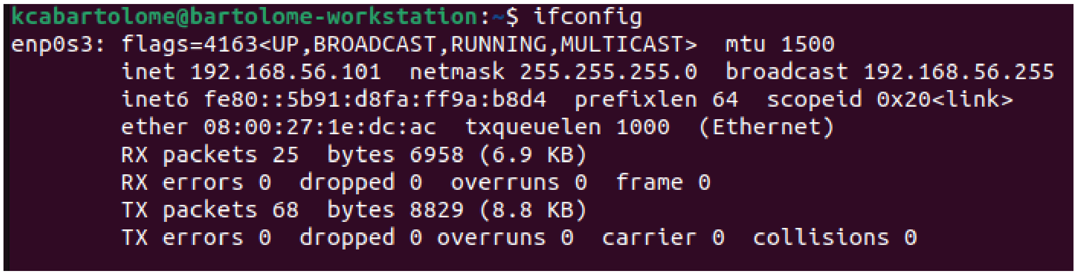
```
kenserver1@bartolome-server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:febd:65aa  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bd:65:aa  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 4235 (4.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1564 (1.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

   1.2 Server 2 IP address: 192.168.56.103



```
kenserver2@bartolome-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::a00:27ff:fe0d:c8d2  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0d:c8:d2  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 4235 (4.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1564 (1.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

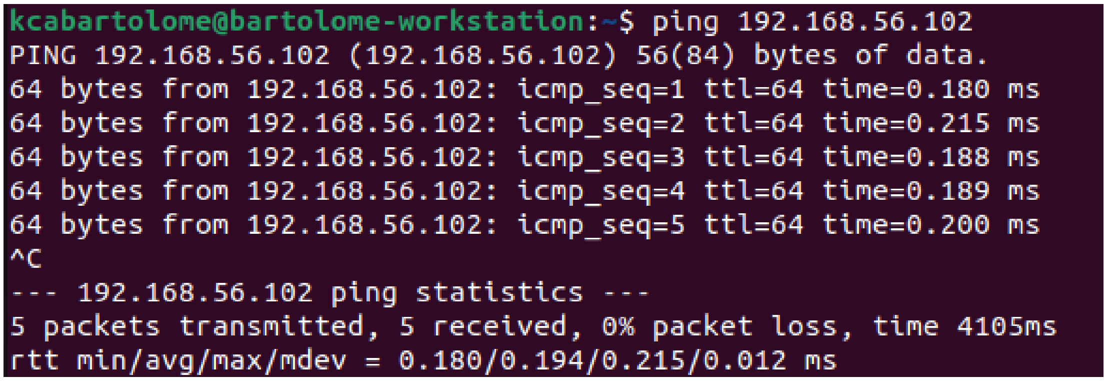
   1.3 Local Machine IP address: 192.168.56.101



```
kcabartolome@bartolome-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::5b91:d8fa:ff9a:b8d4  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1e:dc:ac  txqueuelen 1000  (Ethernet)
        RX packets 25  bytes 6958 (6.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68  bytes 8829 (8.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

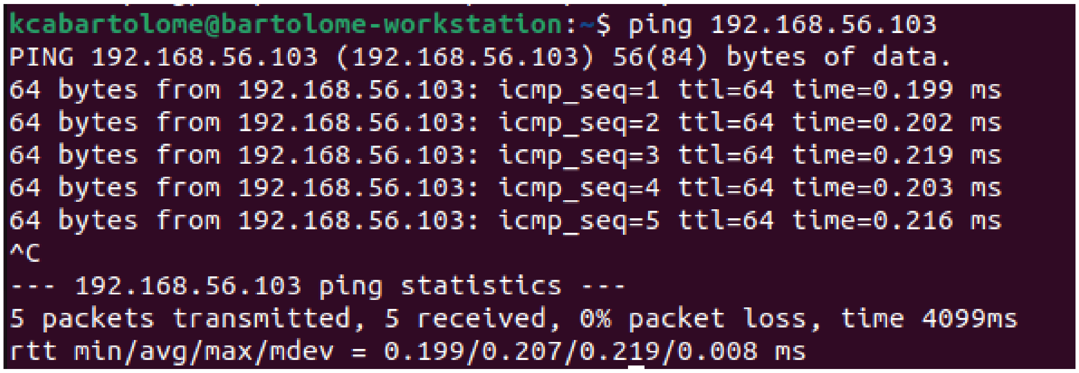
2. Make sure that they can ping each other.

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



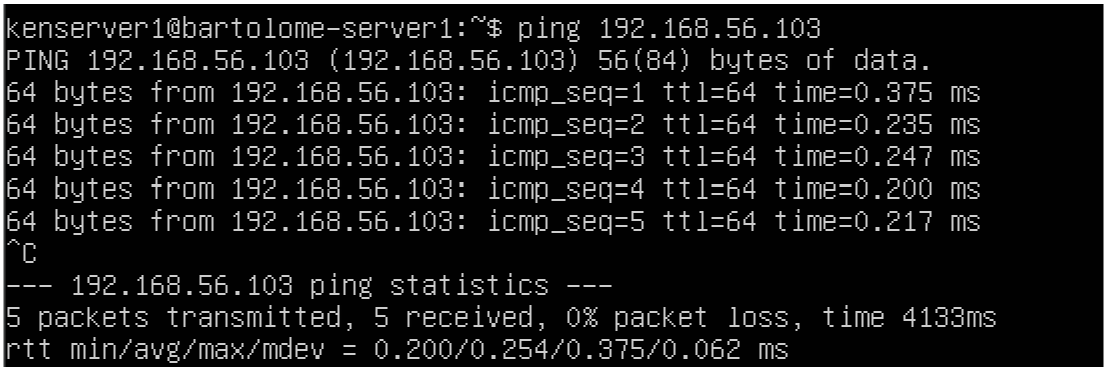
```
kcabartolome@bartolome-workstation:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.180 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.215 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.188 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.189 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.200 ms
^C
--- 192.168.56.102 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4105ms
rtt min/avg/max/mdev = 0.180/0.194/0.215/0.012 ms
```

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



```
kcabartolome@bartolome-workstation:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.199 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.202 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.219 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.203 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.216 ms
^C
--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.199/0.207/0.219/0.008 ms
```

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



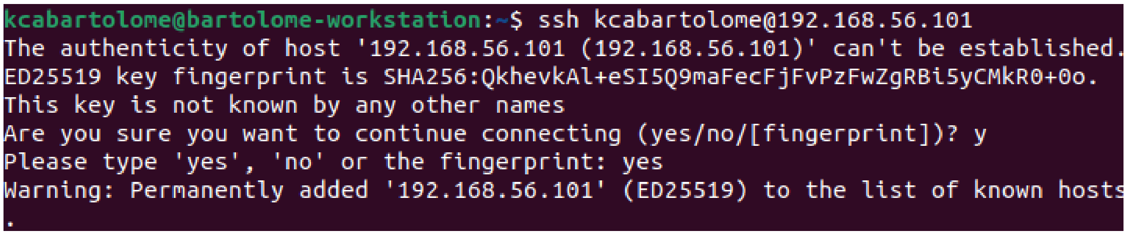
```
kenserver1@bartolome-server1:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.375 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.235 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.247 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.200 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.217 ms
^C
--- 192.168.56.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4133ms
rtt min/avg/max/mdev = 0.200/0.254/0.375/0.062 ms
```

Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

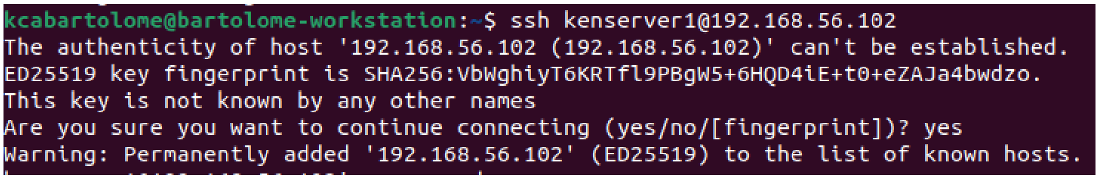
1.1 ssh username@ip_address_server1 for example, *ssh jvtaylar@192.168.56.120*

Workstation:



```
kcabartolome@bartolome-workstation:~$ ssh kcabartolome@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:QkhevkAl+eSI5Q9maFecFjFvPzFwZgRBi5yCMkR0+0o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts
.
```

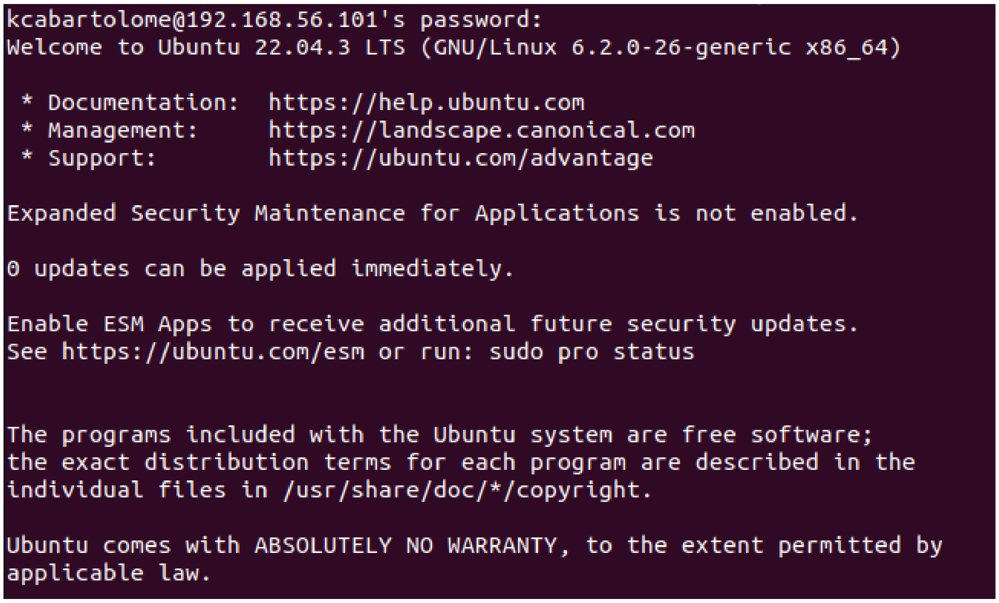
Server1:



```
kcabartolome@bartolome-workstation:~$ ssh kenserver1@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:VbWghiyT6KRTfl9PBgW5+6HQD4iE+t0+eZAJa4bwdzo.
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.
```

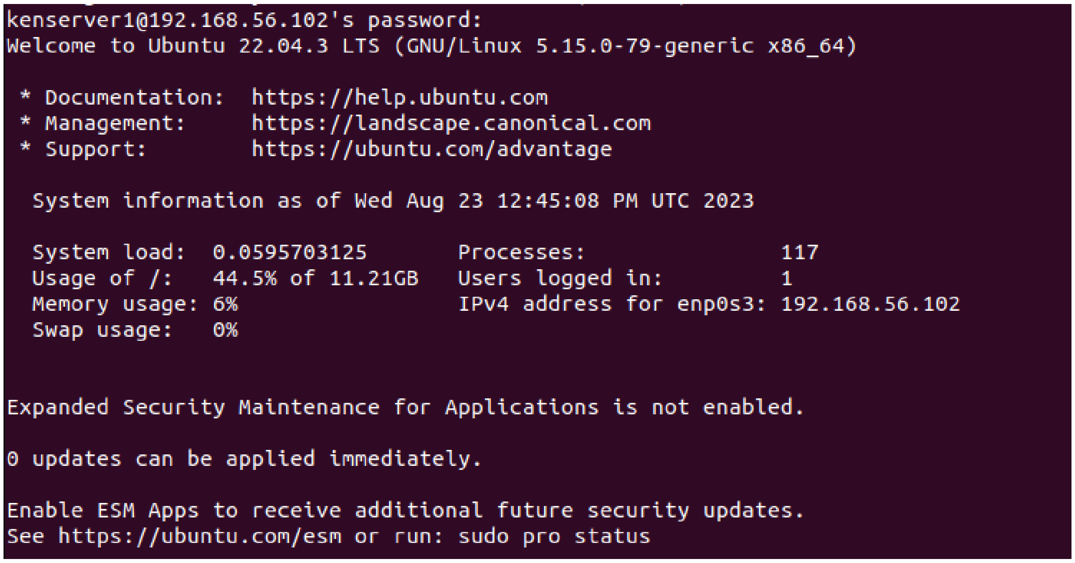
1.2 Enter the password for server 1 when prompted

Workstation:



```
kcabartolome@192.168.56.101'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.
```

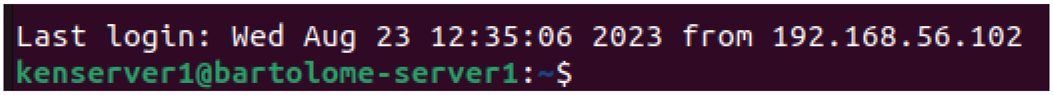
Server1:



```
kenserver1@192.168.56.102'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 Wed Aug 23 12:45:08 PM UTC 2023

  System load:  0.0595703125       Processes:               117
  Usage of /:   44.5% of 11.21GB   Users logged in:         1
  Memory usage: 6%                 IPv4 address for enp0s3: 192.168.56.102
  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
```

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

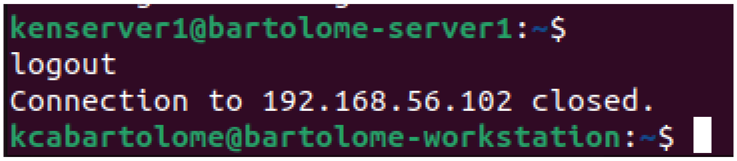
Server1:



```
Last login: Wed Aug 23 12:35:06 2023 from 192.168.56.102
kenserver1@bartolome-server1:~$
```

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

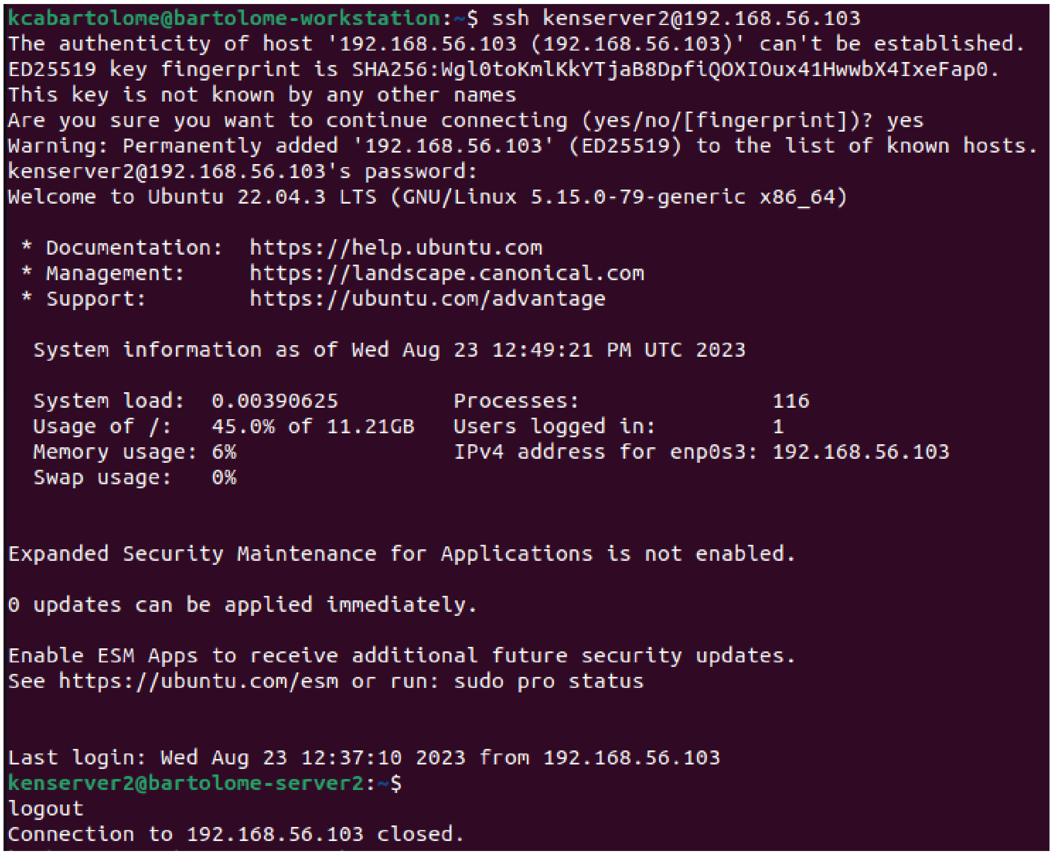
Server1:



```
kenserver1@bartolome-server1:~$
logout
Connection to 192.168.56.102 closed.
kcabartolome@bartolome-workstation:~$ 
```

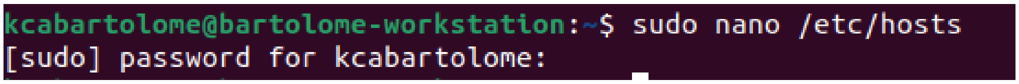
3. Do the same for Server 2.

Server2:



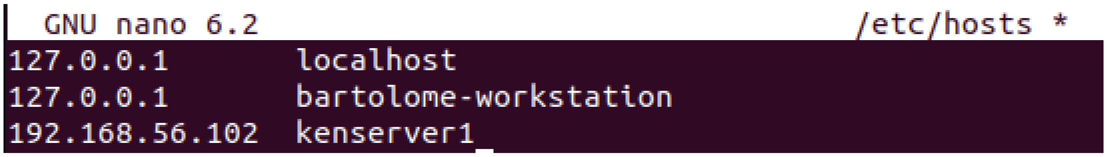
```
kcabartolome@bartolome-workstation:~$ ssh kenserver2@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:Wgl0toKmlKkYTjaB8DpfiQOXIOux41HwwbX4IxeFap0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
kenserver2@192.168.56.103'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 Wed Aug 23 12:49:21 PM UTC 2023

  System load:  0.00390625         Processes:               116
  Usage of /:   45.0% of 11.21GB   Users logged in:         1
  Memory usage: 6%                 IPv4 address for enp0s3: 192.168.56.103
  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: Wed Aug 23 12:37:10 2023 from 192.168.56.103
kenserver2@bartolome-server2:~$
logout
Connection to 192.168.56.103 closed.
```

4. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:



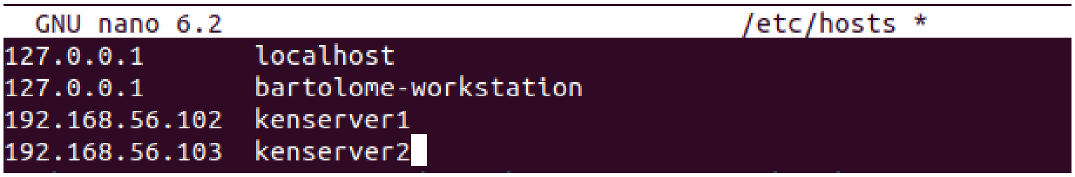
```
kcabartolome@bartolome-workstation:~$ sudo nano /etc/hosts
[sudo] password for kcabartolome:
```

4.1 IP_address server 1 (provide the ip address of server 1 followed by the hostname)



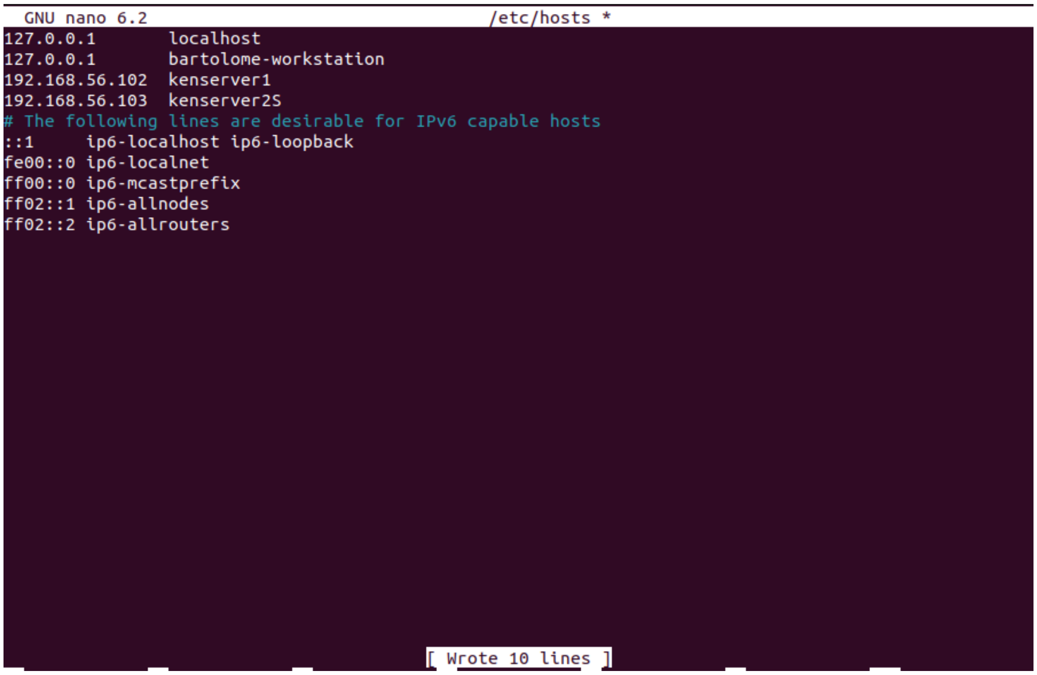
```
  GNU nano 6.2                                        /etc/hosts *
127.0.0.1       localhost
127.0.0.1       bartolome-workstation
192.168.56.102  kenserver1
```

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       bartolome-workstation
192.168.56.102  kenserver1
192.168.56.103  kenserver2
```

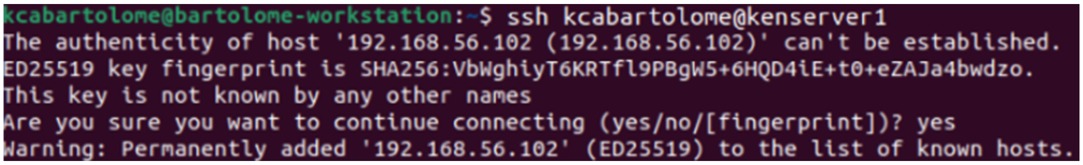
4.3 Save the file and exit.



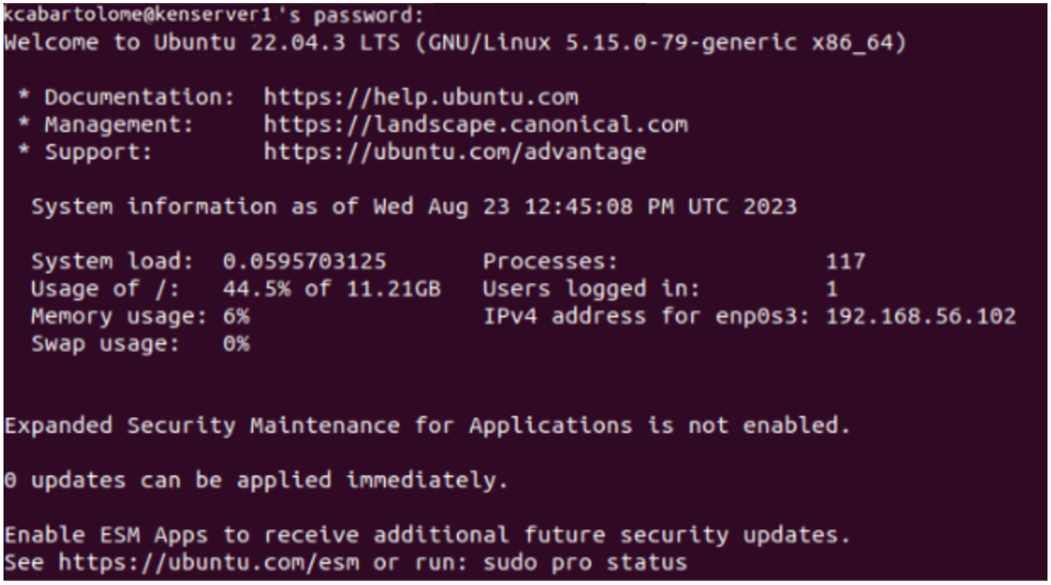
```
  GNU nano 6.2                                /etc/hosts *
127.0.0.1       localhost
127.0.0.1       bartolome-workstation
192.168.56.102  kenserver1
192.168.56.103  kenserver2S
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

                                  [ Wrote 10 lines ]
```

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 jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

Server1:

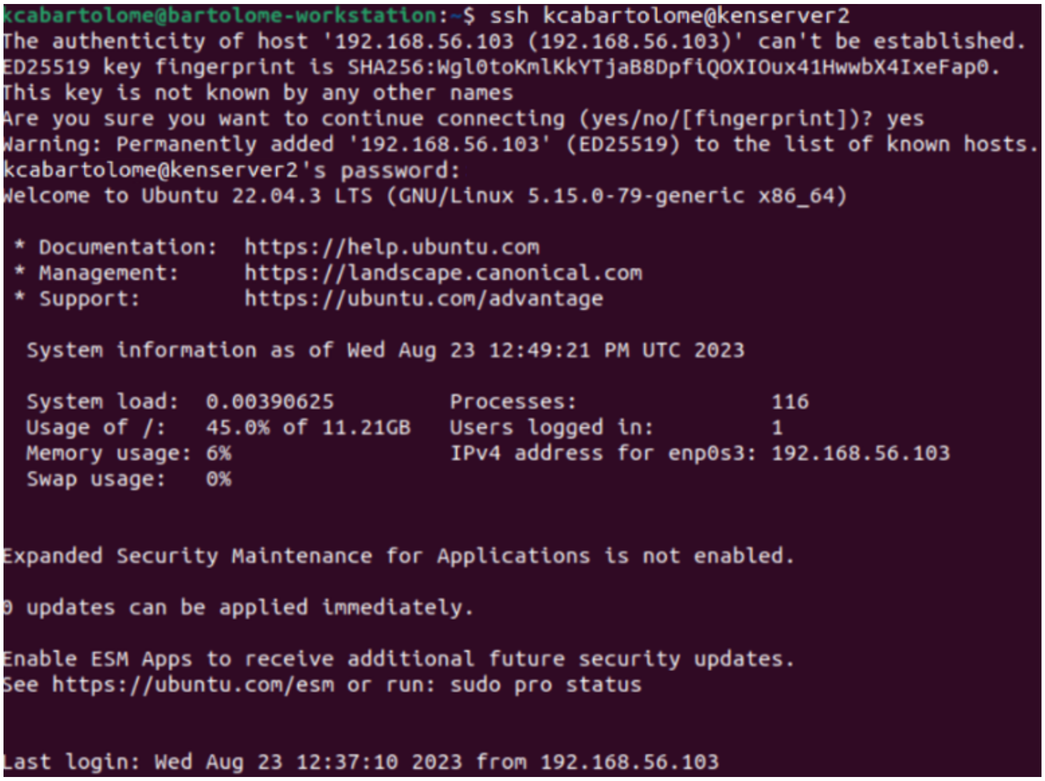


```
kcabartolome@bartolome-workstation:~$ ssh kcabartolome@kenserver1
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:VbWghiyT6KRTfl9PBgW5+6HQD4iE+t0+eZAJa4bwdzo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.
```



```
kcabartolome@kenserver1'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 Wed Aug 23 12:45:08 PM UTC 2023

  System load:  0.0595703125       Processes:               117
  Usage of /:   44.5% of 11.21GB   Users logged in:         1
  Memory usage: 6%                 IPv4 address for enp0s3: 192.168.56.102
  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
```

Server2:



```
kcabartolome@bartolome-workstation:~$ ssh kcabartolome@kenserver2
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:Wgl0toKmlKkYTjaB8DpfiQOXIOux41HwwbX4IxeFap0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
kcabartolome@kenserver2'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 Wed Aug 23 12:49:21 PM UTC 2023

  System load:  0.00390625         Processes:               116
  Usage of /:   45.0% of 11.21GB   Users logged in:         1
  Memory usage: 6%                 IPv4 address for enp0s3: 192.168.56.103
  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: Wed Aug 23 12:37:10 2023 from 192.168.56.103
```

Reflections:

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?

Answer: SSH commands can employ hostnames instead of IP addresses because of DNS resolution. DNS converts domain names that can be read by humans into IP addresses, facilitating better computer communication. This is done by contacting DNS servers, which give the hostnames' associated IP addresses.

2. How secured is SSH?

Answer: Because of its encryption, public key authentication, data integrity safeguards, and programmable security features, SSH (Secure Shell) is extremely secure. Data is encrypted, secure authentication is supported, and data integrity is guaranteed while being transmitted. SSH is typically safe, but to preserve that security over time, it's crucial to keep software updated and adhere to security best practices.