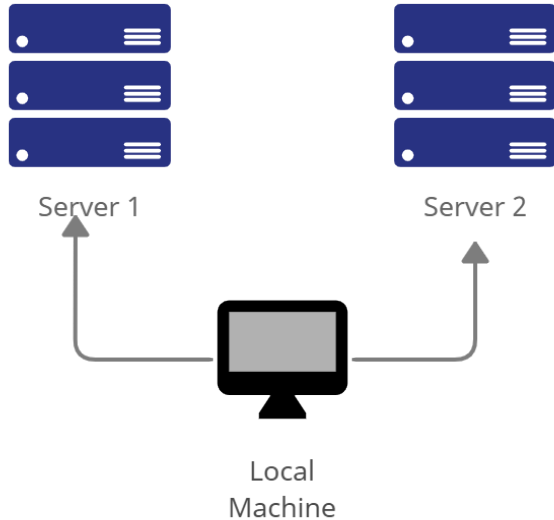
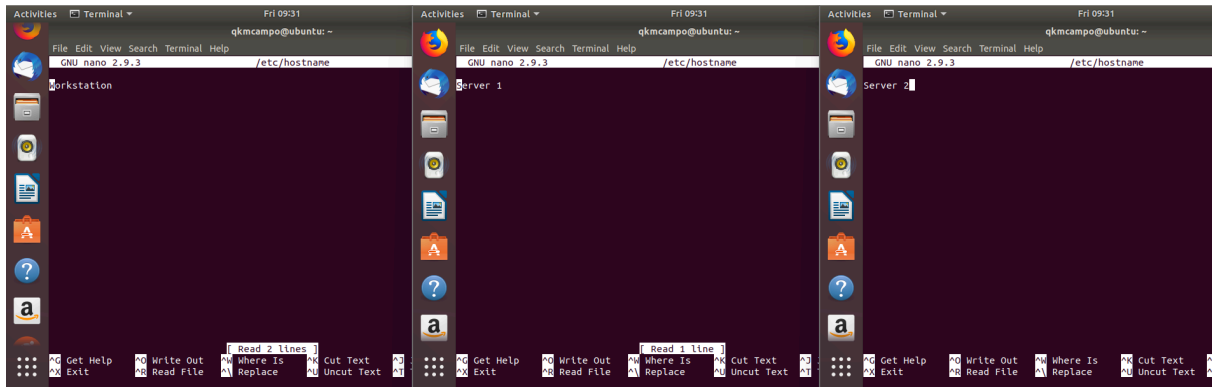
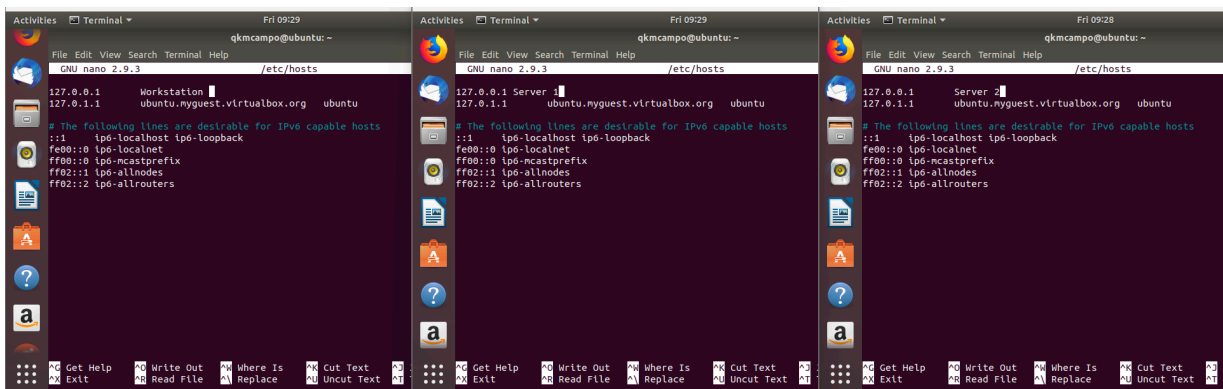


<b>Name: Keneth Campo</b>	<b>Date Performed: August 30, 2024</b>
<b>Course/Section: CPE212- CPE31S21</b>	<b>Date Submitted: August 30, 2024</b>
<b>Instructor: Engr. Robin Valenzuela</b>	<b>Semester and SY: 2024-2025</b>
<b>Activity 1: Configure Network using Virtual Machines</b>	
<b>1. Objectives:</b> 1.1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox 1.2. Set-up a Virtual Network and Test Connectivity of VMs	
<b>2. Discussion:</b>  <b>Network Topology:</b> Assume that you have created the following network topology in Virtual Machines, <i>provide screenshots for each task</i> . (Note: <i>it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine</i> ).	
 <pre> graph TD     LocalMachine[Local Machine] --- Server1[Server 1]     LocalMachine --- Server2[Server 2]   </pre>	
<b>Task 1:</b> Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end. <ol style="list-style-type: none"> <li>1. Change the hostname using the command <i>sudo nano /etc/hostname</i> <ol style="list-style-type: none"> <li>1.1 Use server1 for Server 1</li> <li>1.2 Use server2 for Server 2</li> <li>1.3 Use workstation for the Local Machine</li> </ol> </li> </ol>	

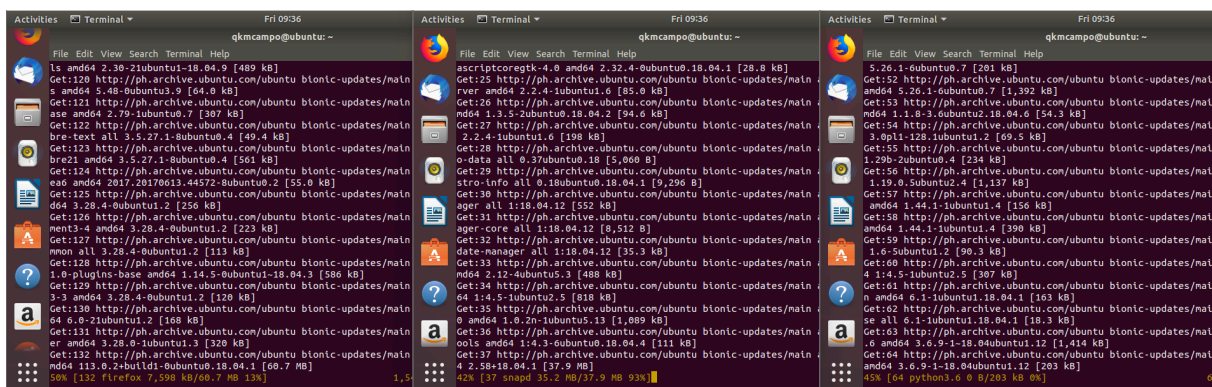


2. Edit the hosts using the command ***sudo nano /etc/hosts***. Edit the second line.
  - 2.1 Type 127.0.0.1 server 1 for Server 1
  - 2.2 Type 127.0.0.1 server 2 for Server 2
  - 2.3 Type 127.0.0.1 workstation for the Local Machine



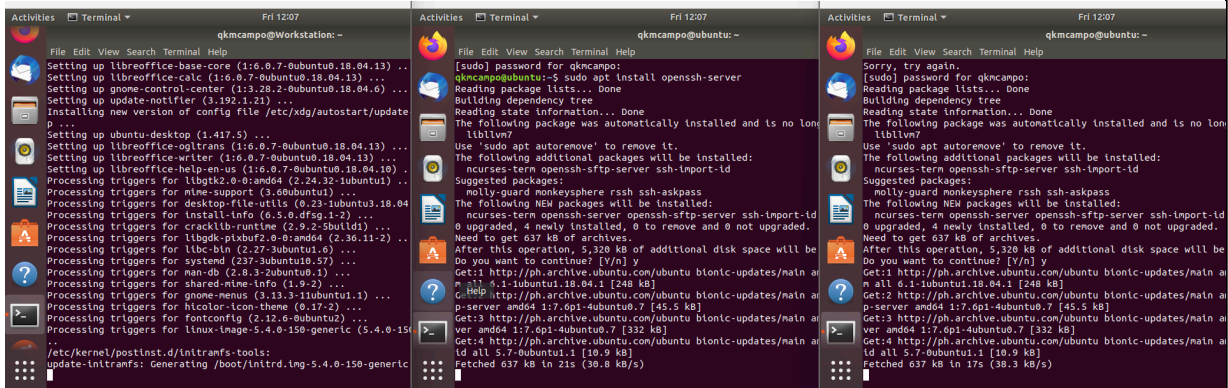
## Task 2: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:

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



***upgrade*** respectively.

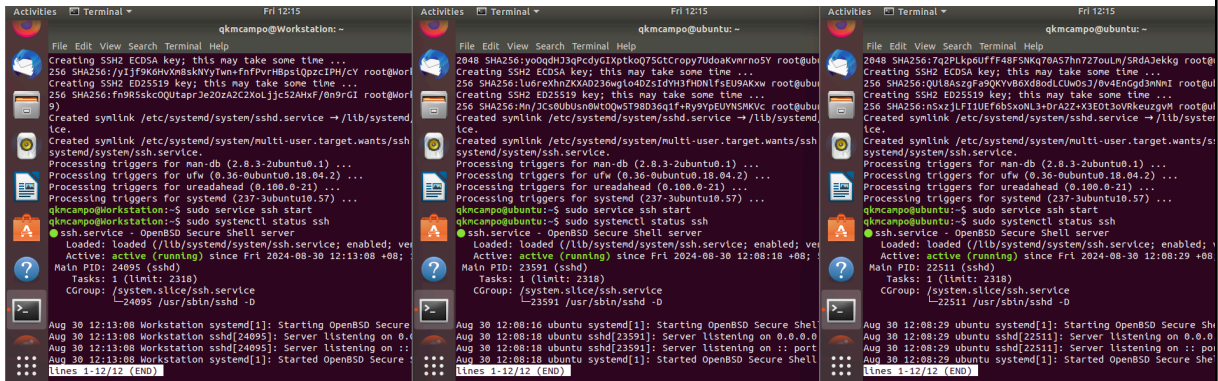
## 2. Install the SSH server using the command *sudo apt install openssh-server*.



```
qkmcampo@Workstation:~$ sudo apt install openssh-server
[sudo] password for qkmcampo:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
libblivet1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
molly-guard monkeysphere rssh 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 0 not upgraded.
Need to get 637 kB of archives.
After this operation, 5,320 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu blonic-updates/main amd64 ncurses-term 6.4-2ubuntu1 [248 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu blonic-updates/main amd64 openssh-sftp-server 9.9p1-3ubuntu1 [45.5 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu blonic-updates/main amd64 openssh-server 9.9p1-3ubuntu1 [45.5 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu blonic-updates/main amd64 ssh-import-id 5.10-0ubuntu1 [10.9 kB]
Fetched 637 kB in 21s (30.8 kB/s)
```

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

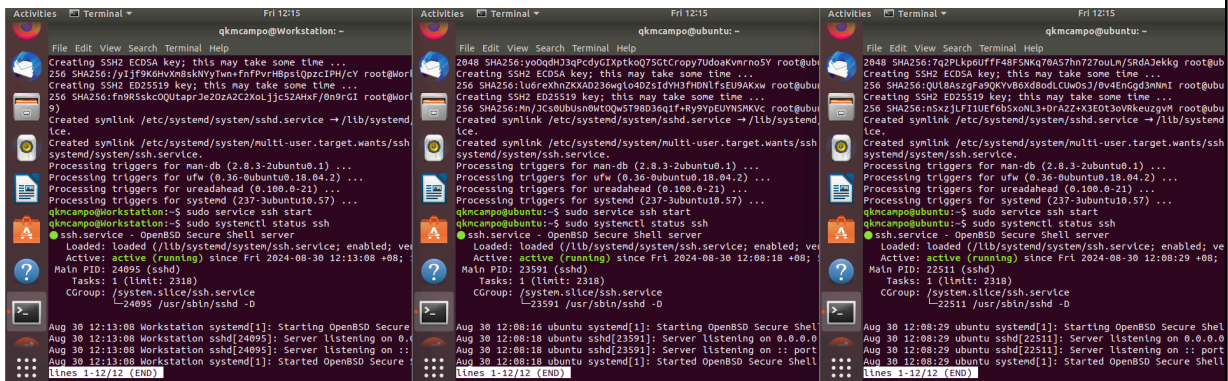
### 3.1 *sudo service ssh start*



```
qkmcampo@Workstation:~$ sudo service ssh start
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
Processing triggers for man-db (2.8.3-2ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.57) ...
qkmcampo@Workstation:~$ sudo systemctl status ssh
● ssh.service - OpenSSH Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-08-30 12:13:08 +08; 1min 23s ago
     Main PID: 24895 (sshd)
       Tasks: 1 (limit: 2318)
      CGroup: /system.slice/ssh.service
              └─24895 /usr/sbin/sshd -D

Aug 30 12:13:08 Workstation systemd[1]: Starting OpenSSH Secure Shell server: sshd(24895): Server listening on 0.0.0.0 port 22.
Aug 30 12:13:08 Workstation sshd[24895]: Server listening on 0.0.0.0 port 22.
Aug 30 12:13:08 Workstation systemd[1]: Started OpenSSH Secure Shell server: sshd(24895): Server listening on 0.0.0.0 port 22.
lines 1-12/12 (END)
```

### 3.2 *sudo systemctl status ssh*



```
qkmcampo@Workstation:~$ sudo systemctl status ssh
● ssh.service - OpenSSH Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-08-30 12:13:08 +08; 1min 23s ago
     Main PID: 24895 (sshd)
       Tasks: 1 (limit: 2318)
      CGroup: /system.slice/ssh.service
              └─24895 /usr/sbin/sshd -D

Aug 30 12:13:08 Workstation systemd[1]: Starting OpenSSH Secure Shell server: sshd(24895): Server listening on 0.0.0.0 port 22.
Aug 30 12:13:08 Workstation sshd[24895]: Server listening on 0.0.0.0 port 22.
Aug 30 12:13:08 Workstation systemd[1]: Started OpenSSH Secure Shell server: sshd(24895): Server listening on 0.0.0.0 port 22.
lines 1-12/12 (END)
```

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

#### 4.1 *sudo ufw allow ssh*

```
qkncampo@Workstation:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
qkncampo@Workstation:~$
```

```
Aug 30 12:08:18 ubuntu systemd[1]: Started OpenSSH Secure Shell.
qkncampo@ubuntu:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
qkncampo@ubuntu:~$
```

```
Aug 30 12:08:29 ubuntu systemd[1]: Started OpenSSH Secure Shell.
qkncampo@ubuntu:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
qkncampo@ubuntu:~$
```

#### 4.2 *sudo ufw enable*

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

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

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

#### 4.3 *sudo ufw status*

```
qkncampo@Workstation:~$ sudo ufw status
Status: active

To Action From
--
22/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
qkncampo@Workstation:~$
```

```
qkncampo@ubuntu:~$ sudo ufw status
Status: active

To Action From
--
22/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
qkncampo@ubuntu:~$
```

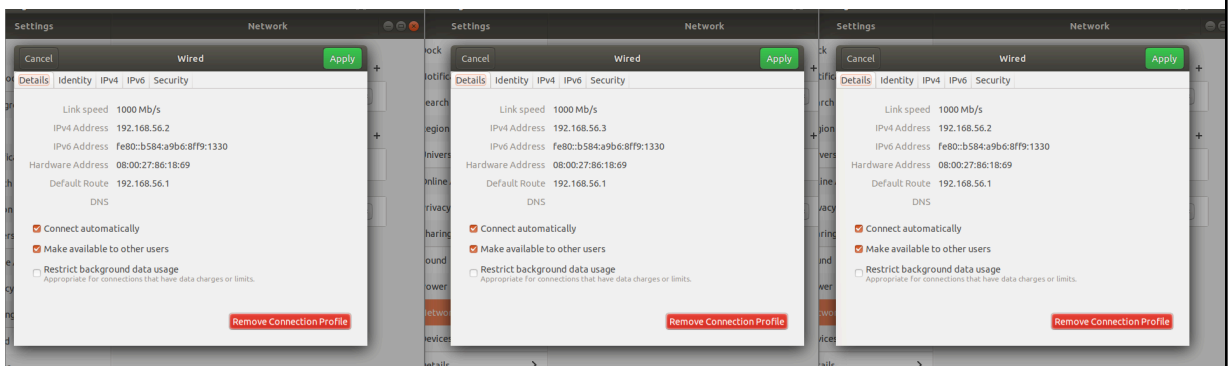
```
qkncampo@ubuntu:~$ sudo ufw status
Status: active

To Action From
--
22/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
qkncampo@ubuntu:~$
```

**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.\_\_\_\_
- 1.2 Server 2 IP address: 192.168.56.\_\_\_\_
- 1.3 Server 3 IP address: 192.168.56.\_\_\_\_

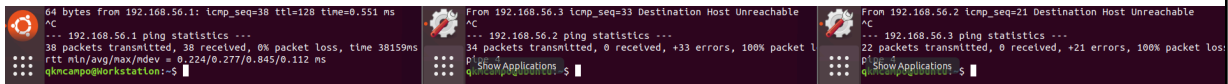


2. Make sure that they can ping each other.

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

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

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

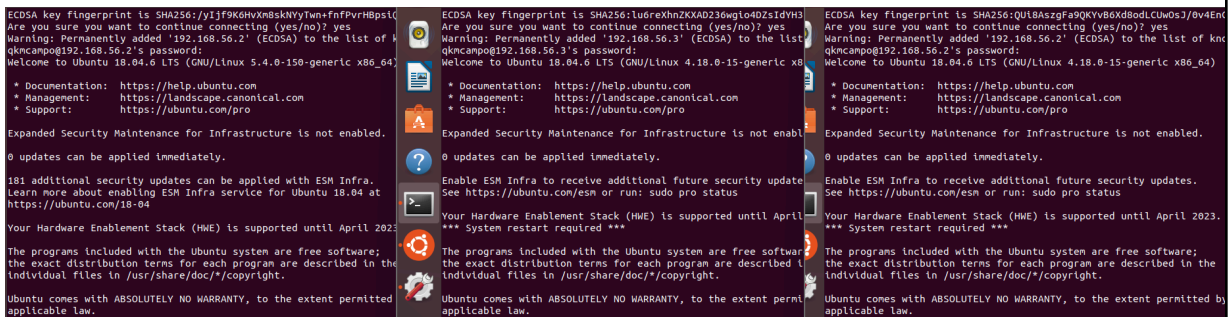


### Task 4: Verify SSH connectivity on Server 1, Server 2, and Local Machine.

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

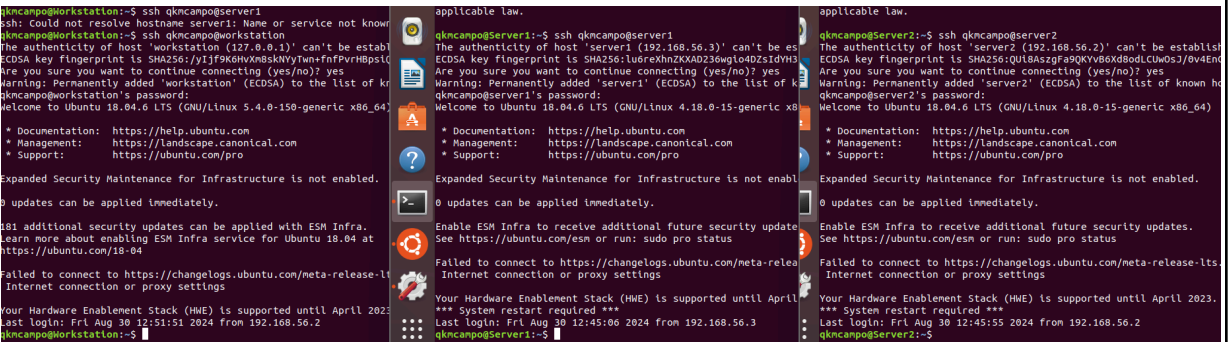
1.1 `ssh username@ip_address_server1` for example, `ssh jvtaylor@192.168.56.120`

1.2 Enter the password for server 1 when prompted

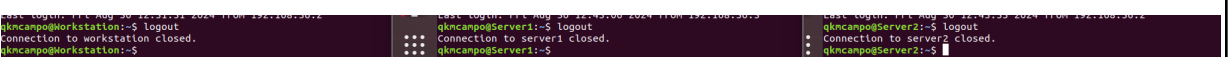


1.3 Verify that you are in server 1. The user should be in this format `user@server1`.

For example, `jvtaylor@server1`



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



3. Do the same for Server 2.

4. Edit the hosts of the Local Machine by issuing the command `sudo nano /etc/hosts`. Below all texts type the following:
  - 4.1 `IP_address server 1` (provide the ip address of server 1 followed by the hostname)
  - 4.2 `IP_address server 2` (provide the ip address of server 2 followed by the hostname)
  - 4.3 Save the file and exit.
5. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do `ssh jvtaylor@server1`. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.

**Reflections:**

Answer the following:

1. How are we able to use the hostname instead of IP address in SSH commands?
  - The hostname in SSH commands by making changes to `/etc/hosts` file on your computer.
2. How secured is SSH?
  - Yes SSH is secure enough , SSH is encrypted theoretically. The security measures SSH an option for establishing remote connections.

Conclusion:

Therefore I learned how to change the server name and hostname. change the ip address and ping the ip address is quite hard since the its very long to update and upgrade we used our time to update and upgrade.