

# ASSIGNMENT-1

## BY NISHITHA TANUKUNURI (A20537907)

---

### SETUP VM, LINUX AND BASIC TESTING

1. Read Oracle VirtualBox White Paper

The image shows a screenshot of a white paper page from Oracle. At the top left is the Oracle logo. Below it, the title "Oracle Virtualbox" is displayed in large, bold, black font. To the right of the title is a photograph of a server room with multiple server racks. Below the title, there's a large, light gray rectangular area containing the main content of the white paper. In the top right corner of this area, the Oracle VM VirtualBox logo is visible. The main content area starts with the text "An Oracle White Paper March 2016" followed by "Oracle VM VirtualBox 5.0 Overview". On the right side of the content area, there is a descriptive text block and a bulleted list of features. At the bottom right of the content area is a blue button labeled "Download White Paper" with a white arrow icon.

ORACLE

## Oracle Virtualbox

An Oracle White Paper  
March 2016

Oracle VM VirtualBox 5.0 Overview

Oracle VM VirtualBox is for Everyone Oracle VM VirtualBox runs across Windows, Mac OSX, Linux and Solaris allowing users to run multiple different guest operating systems. Download this paper to:

- Understand the key features and typical use cases of VirtualBox 5.0
- Discover why Oracle VM VirtualBox is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms from one machine
- Learn about Oracle VM VirtualBox 5.0 new innovative features that deliver excellent performance and flexibility to a wide range of supported guest operating system

[Download White Paper](#)

---

ORACLE

# Oracle Virtualbox



Thank you Nishitha Tanukunuri

Thank you for your interest in the VM Virtual Box whitepaper.

You'll gain insight into VirtualBox and the latest 5.0 release.

[Download White Paper](#)



## Talk to a Sales Expert

[Chat Now](#) or Call 1.800.633.0738

## Stay Connected



## You might also be interested in:

Video: Oracle VM Virtualbox Video Demo

Oracle VM Virtualbox enhances user's ability to create and deploy virtual machines nearly everywhere, upload and download to the cloud, and review and make changes offline.



ORACLE

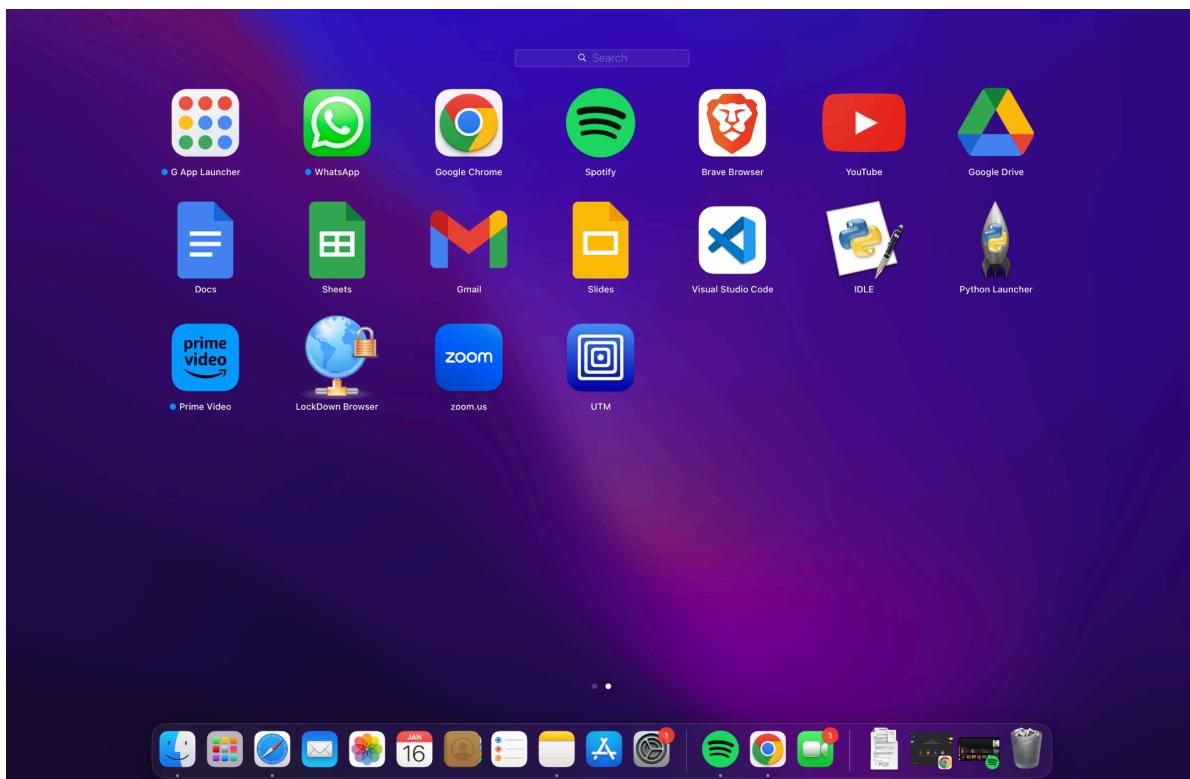
## Oracle VM VirtualBox Overview

An Oracle White Paper

June, 2021, Version 2.0  
Copyright © 2021, Oracle and/or its affiliates  
Public

---

## 2. Download UTM



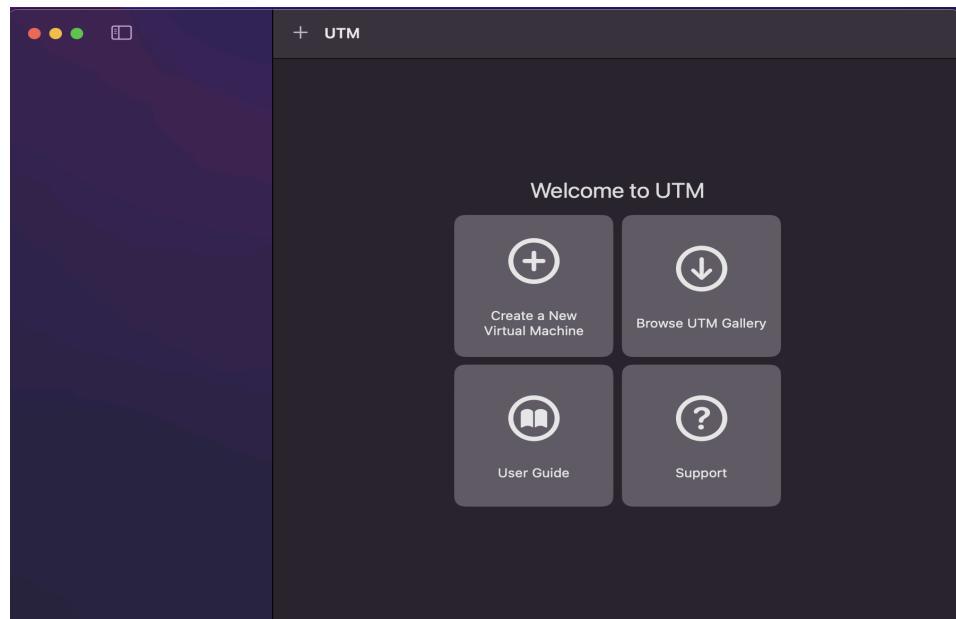
---

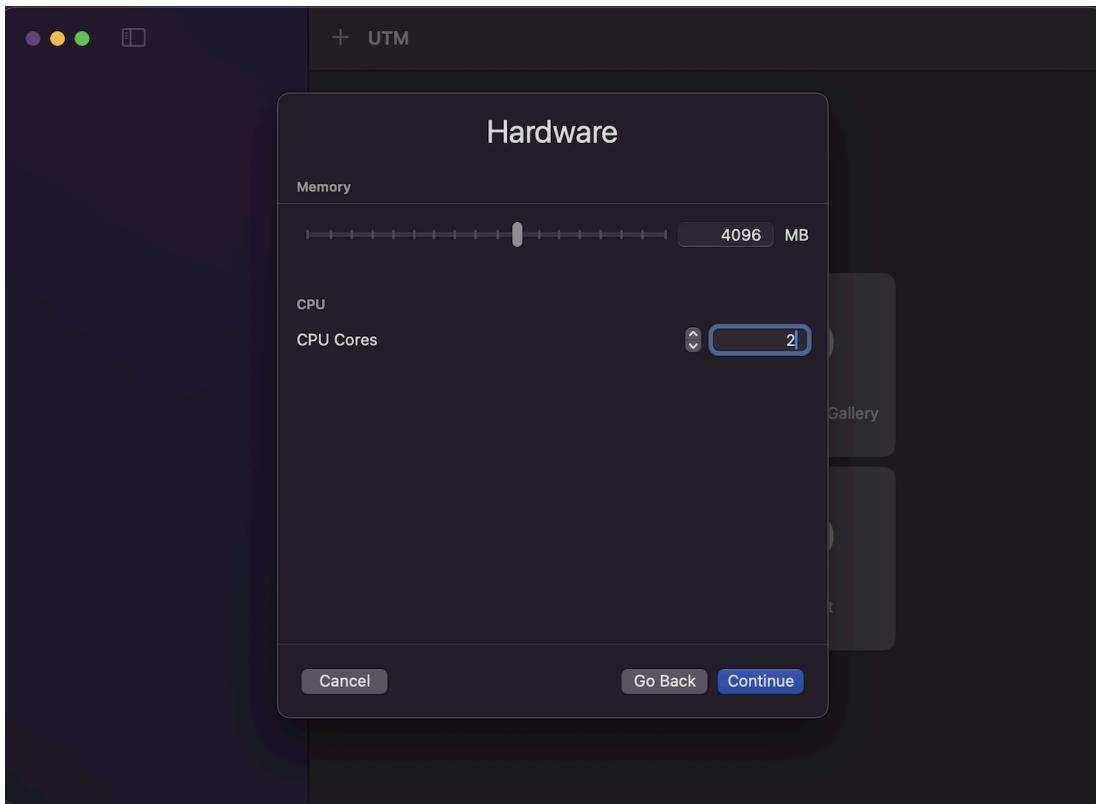
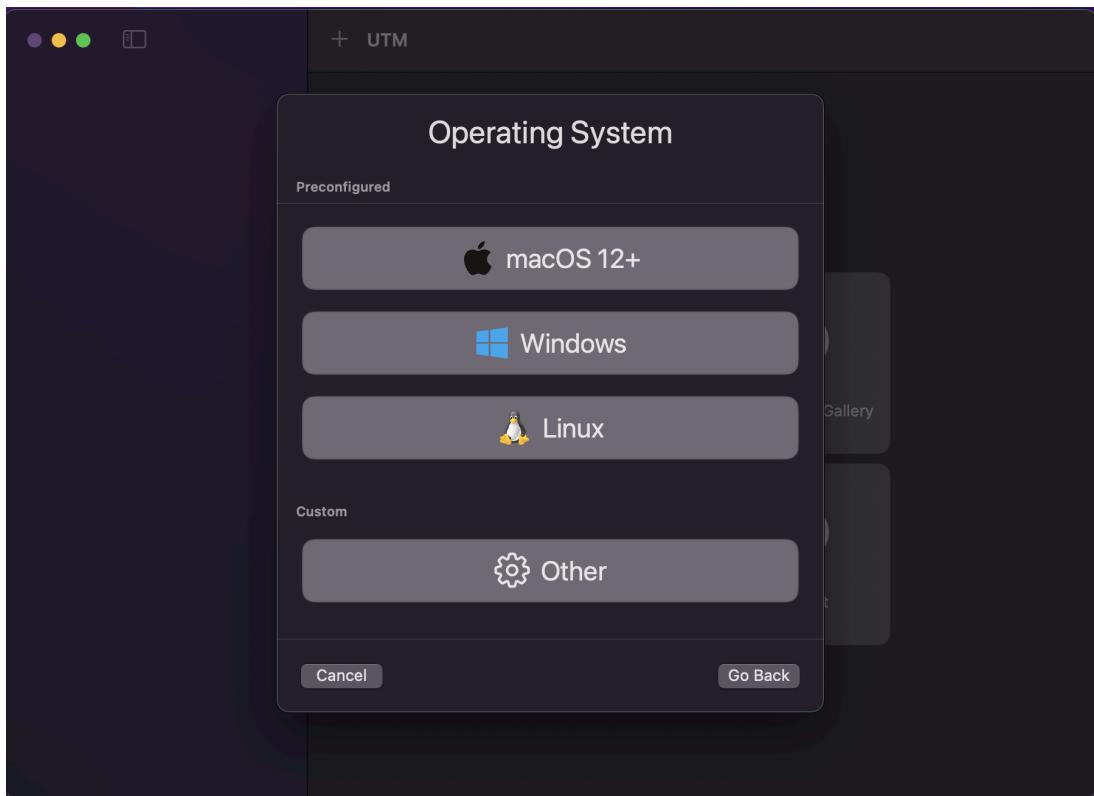
### 3. Download Ubuntu 22.04 Linux ISO Image

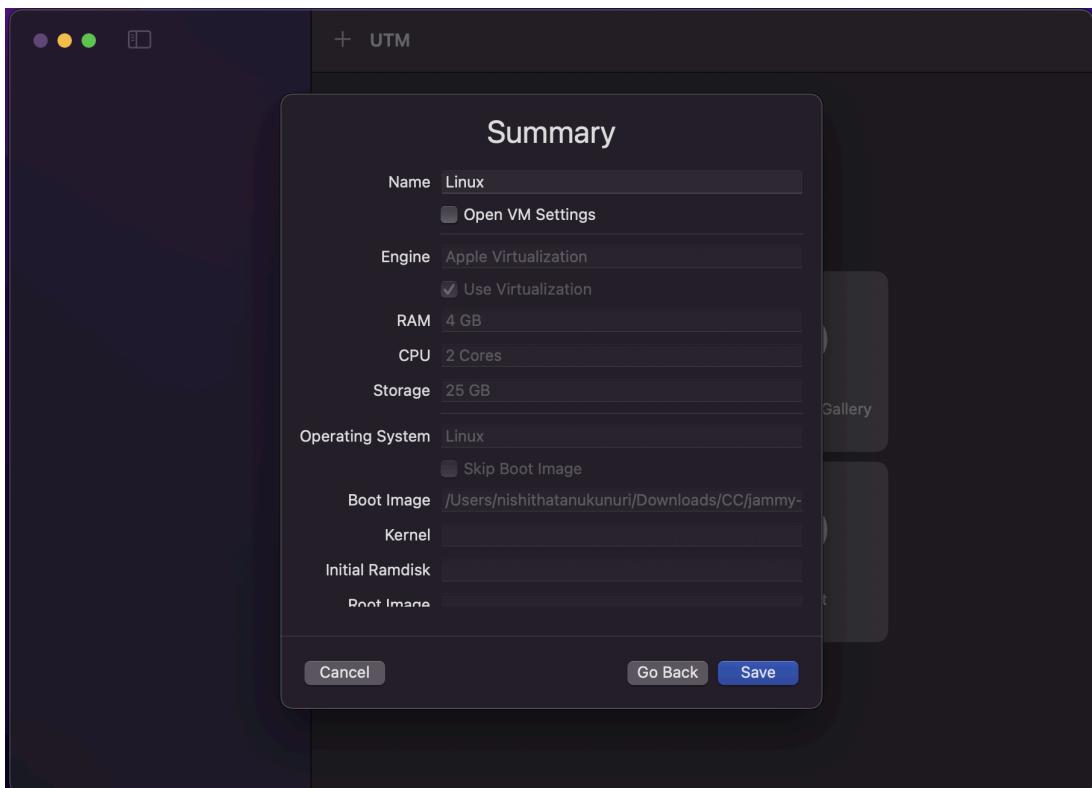
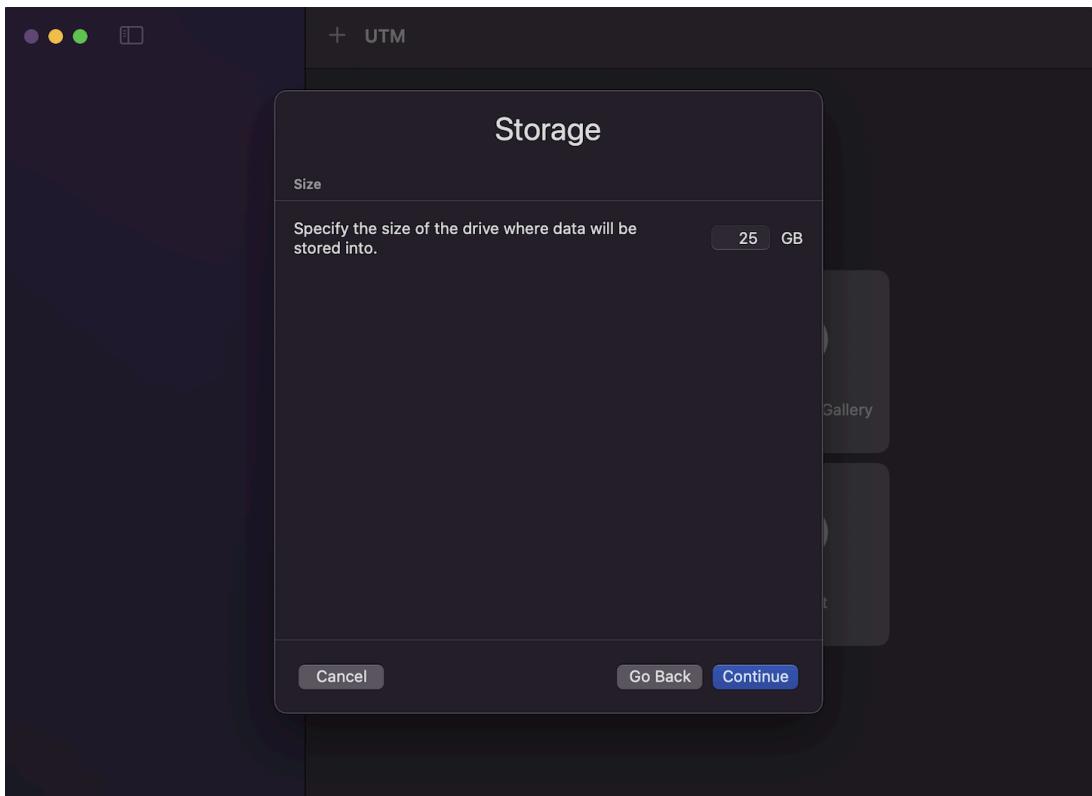


The screenshot shows the Ubuntu 22.04.3 LTS (Jammy Jellyfish) download page. At the top, there's a header with the Ubuntu logo and the text "ubuntu releases". Below the header, the title "Ubuntu 22.04.3 LTS (Jammy Jellyfish)" is displayed. A large orange banner with the text "Select an image" is visible. Underneath, two options are shown: "Desktop image" and "Server install image". Each option has a detailed description and a "64-bit PC (AMD64) desktop image" or "64-bit PC (AMD64) server install image" link. At the bottom of the page, a note states: "A full list of available files, including BitTorrent files, can be found below."

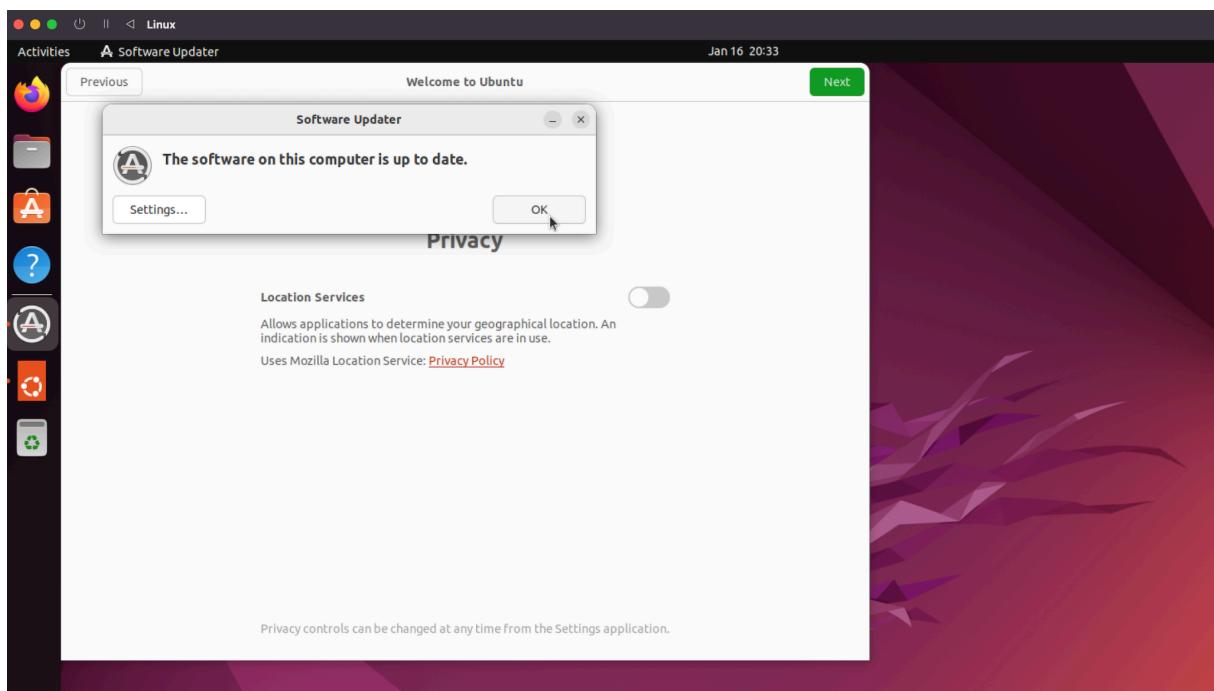
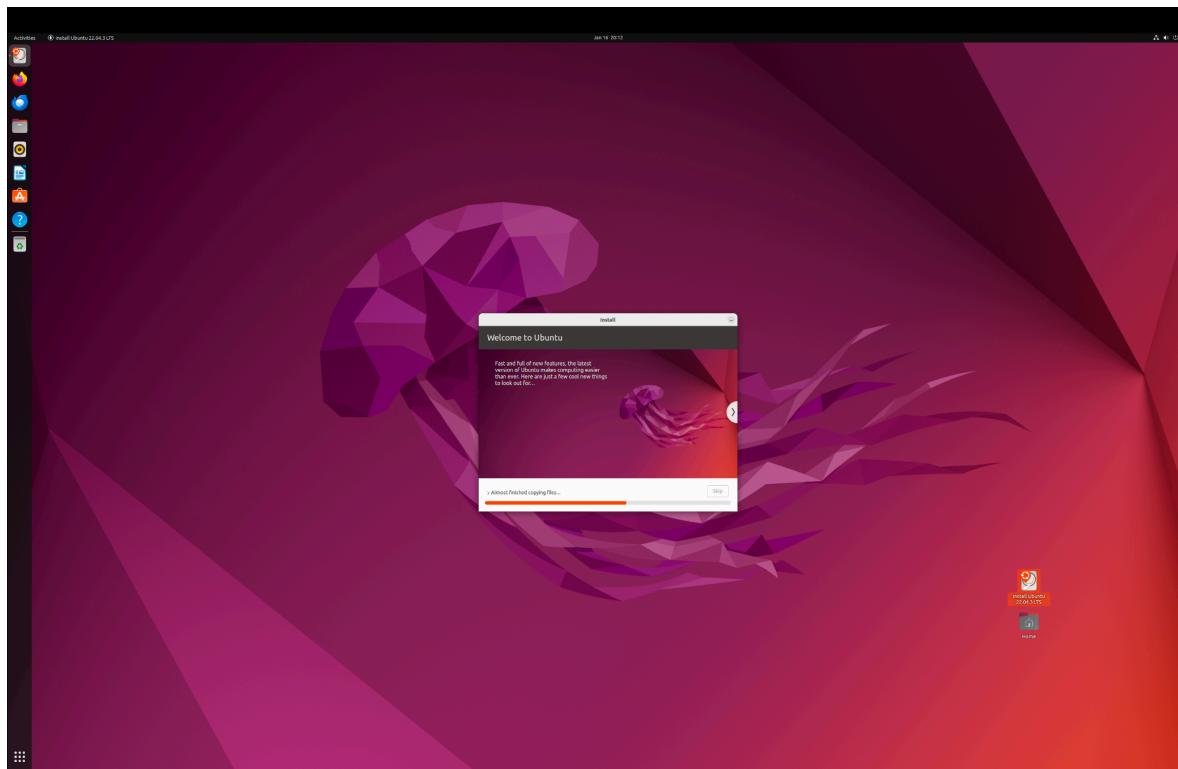
### 4. Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with Bridged Adapter







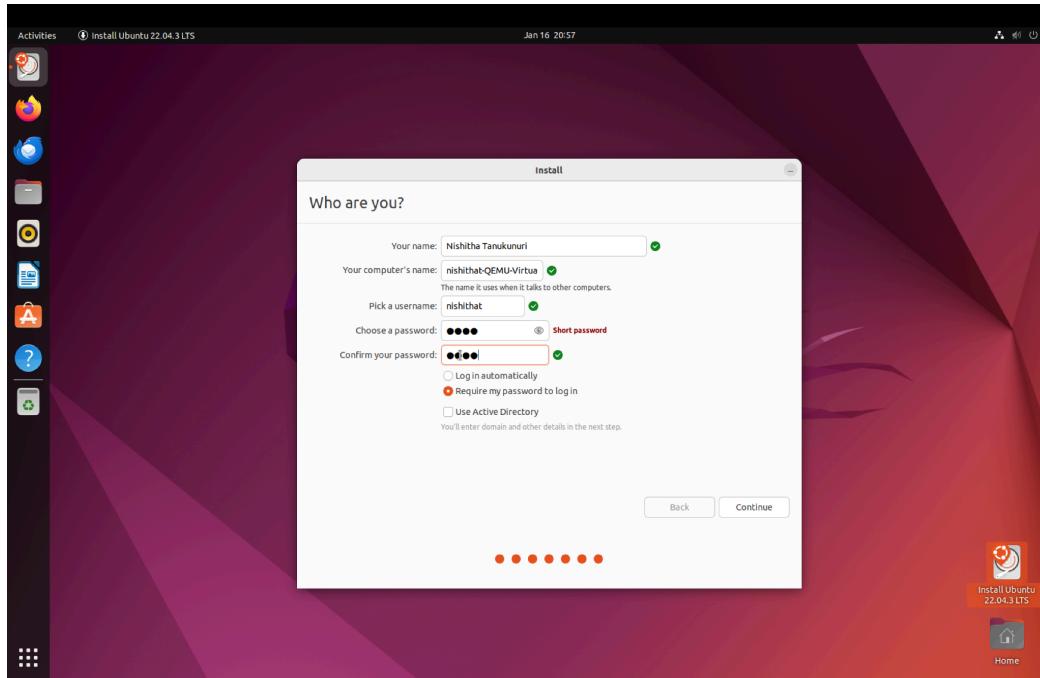
## 5. Install Linux from the ISO image



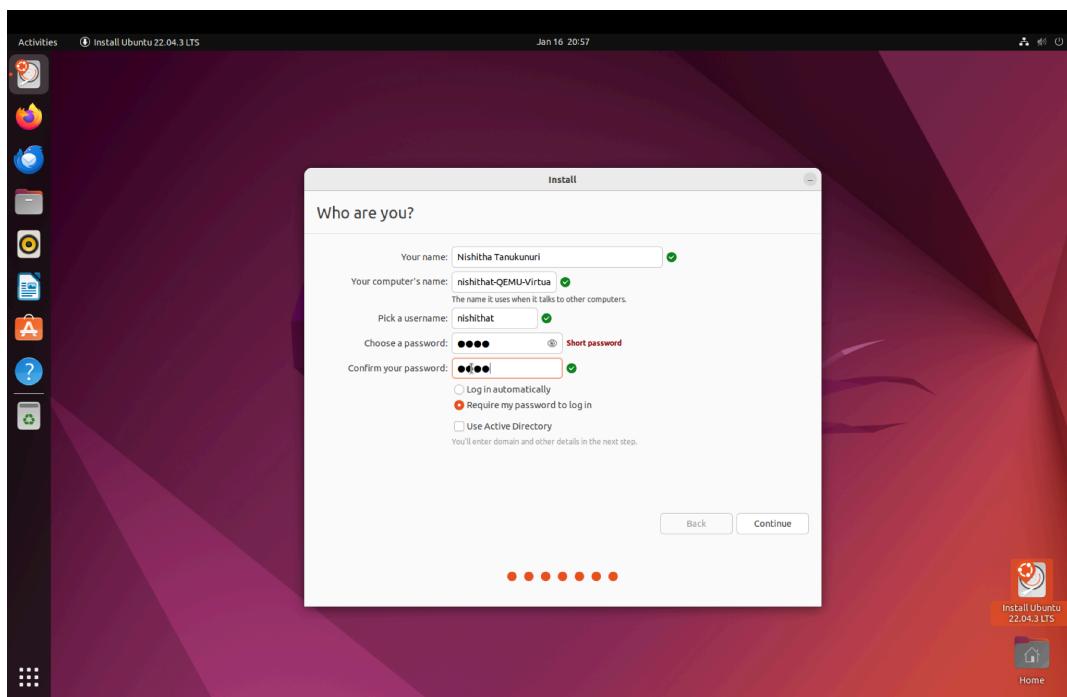
---

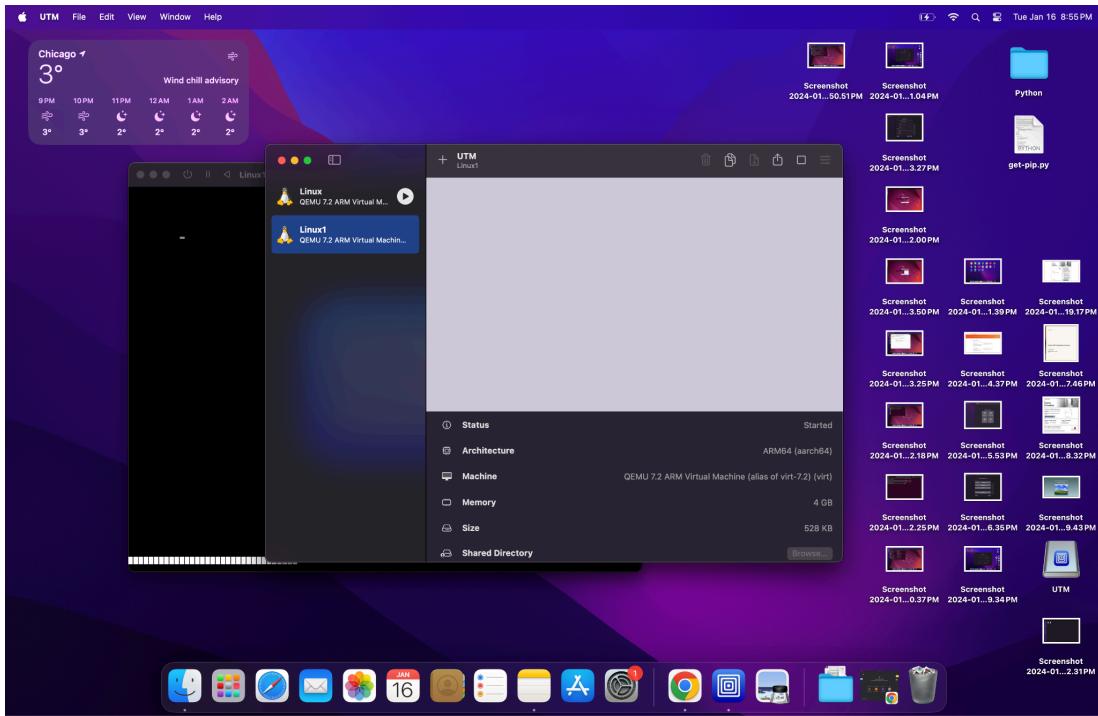
## 6. Create a user ID and password

**For VM 1:**



**For VM 2:**





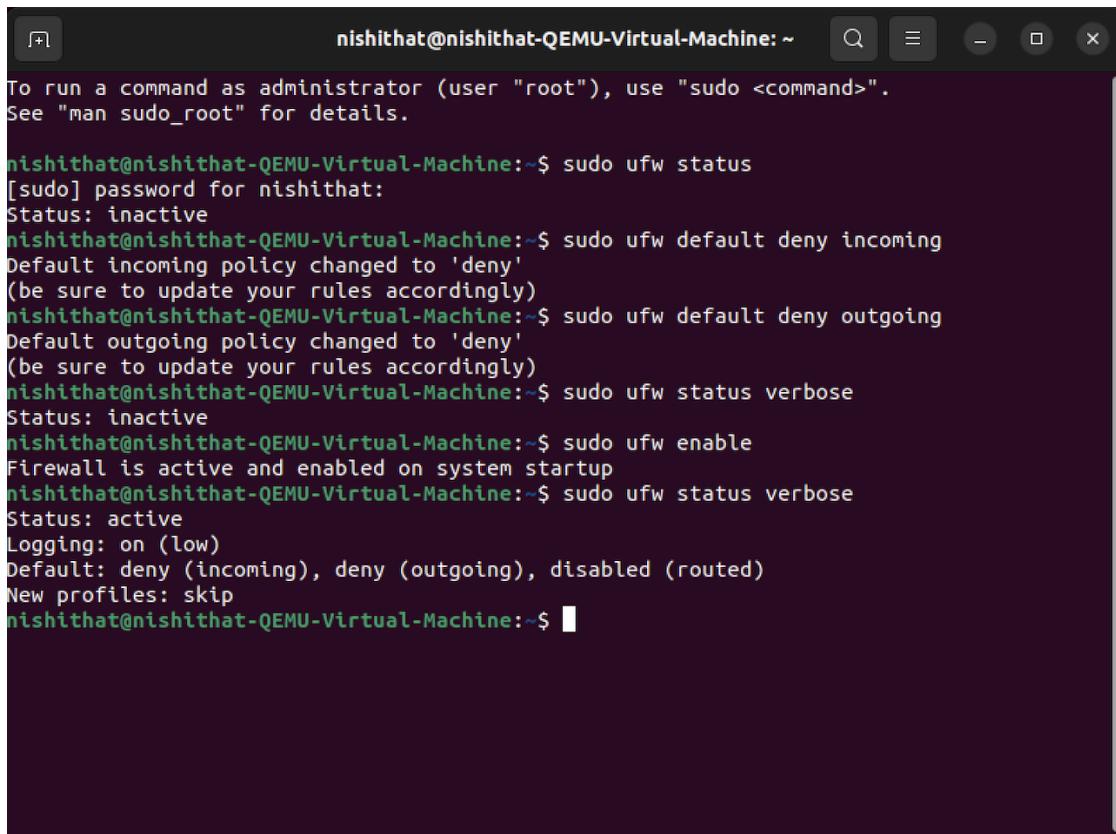
## 7. Turn on Firewall and block all ports

**For VM 1:**

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw status
Status: inactive
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw status
Status: active
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw disable
Firewall stopped and disabled on system startup
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw enable verbose
ERROR: Invalid syntax
```

---

### For VM 2:



The screenshot shows a terminal window with the title bar "nishithat@nishithat-QEMU-Virtual-Machine: ~". The terminal displays the following command-line session:

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw status
[sudo] password for nishithat:
Status: inactive

[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)

[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw status verbose
Status: inactive

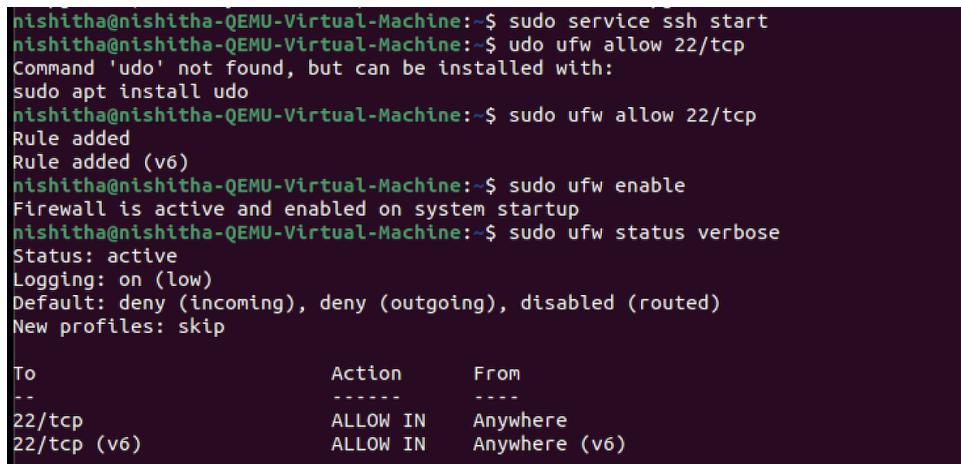
[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw enable
Firewall is active and enabled on system startup

[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

[nishithat@nishithat-QEMU-Virtual-Machine:~]$
```

8. Enable SSH access to your new Linux installation; open SSH port in firewall

### For VM 1:



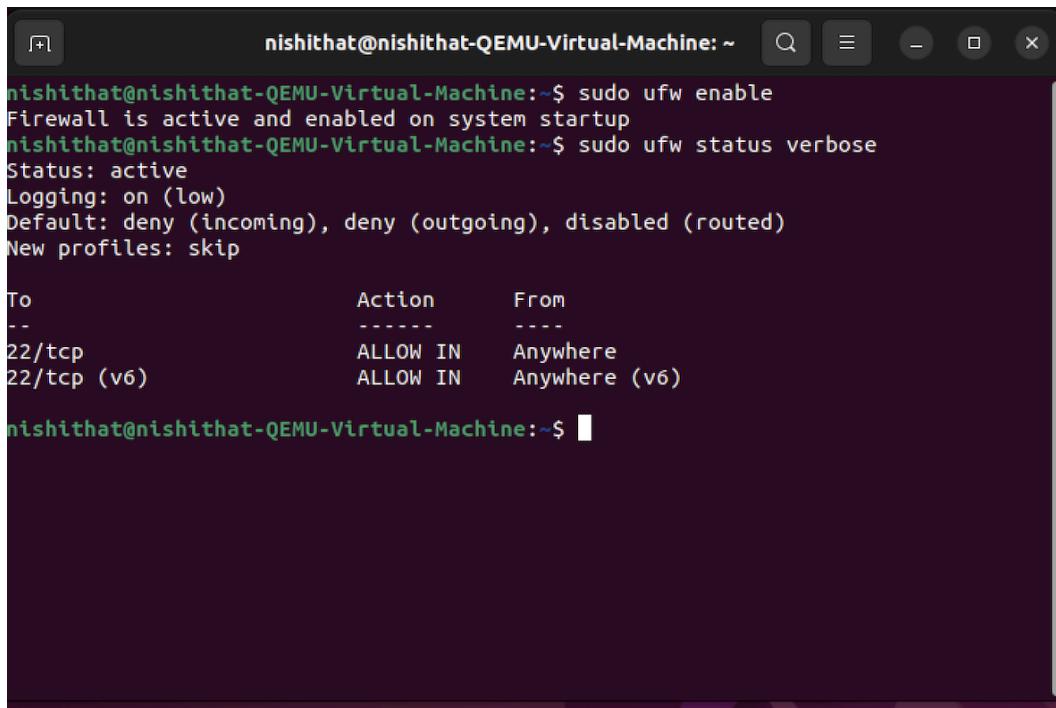
The screenshot shows a terminal window with the title bar "nishitha@nishitha-QEMU-Virtual-Machine: ~". The terminal displays the following command-line session:

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo service ssh start
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw allow 22/tcp
Command 'udo' not found, but can be installed with:
sudo apt install udo
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

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

---

### For VM 2:



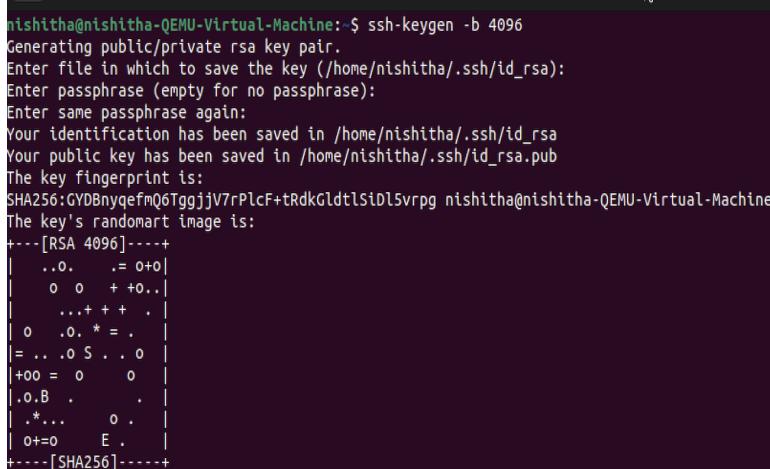
```
nishithat@nishithat-QEMU-Virtual-Machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
nishithat@nishithat-QEMU-Virtual-Machine:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

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

nishithat@nishithat-QEMU-Virtual-Machine:~$
```

9. Create private/public keys and install them properly in both of your new VMs

### For VM 1:



```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nishitha/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nishitha/.ssh/id_rsa
Your public key has been saved in /home/nishitha/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GYDBnyqefmQ6TggjjV7rPlcF+tRdkGldtlSiDl5vrpg nishitha@nishitha-QEMU-Virtual-Machine
The key's randomart image is:
+---[RSA 4096]---+
| ..o. . = o+o|
| o o + +o..|
| ...+ + + . |
| o .o * = . |
| = ... o S . . o |
| +oo = o o |
| .o.B . . |
| .*... o . |
| o+=o E . |
+---[SHA256]-----
```

---

### For VM 2:

```
nishithat@nishithat-QEMU-Virtual-Machine:~$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nishithat/.ssh/id_rsa):
/home/nishithat/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nishithat/.ssh/id_rsa
Your public key has been saved in /home/nishithat/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0IPGlfFcoACKdNofnkR4tzfYhvQwLTUR6JUdgqIBruI nishithat@nishithat-QEMU-Virtual-Machine
The key's randomart image is:
+---[RSA 4096]---+
| oooo.o=***... |
| .+o.*.%..+.. |
| oo .+o& %     |
| . .+ooB B     |
| o   +. S .    |
| o             |
| E             |
|               |
+--- [SHA256] ---+
```

- 10.** Test that you can connect remotely to your VMs with your keys, from one VM to the other VM

### For VM 1:

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ssh nishithat@192.168.64.7
nisithat@192.168.64.7's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

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.
```

---

**For VM 2:**

```
nishithat@nishithat-QEMU-Virtual-Machine:~$ ssh nishitha@192.168.64.6
The authenticity of host '192.168.64.6 (192.168.64.6)' can't be established.
ED25519 key fingerprint is SHA256:yuhXsWkC0KyS32Rnt7Mikpv00QQMJF5+hgPW1CbckAs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.64.6' (ED25519) to the list of known hosts
nishitha@192.168.64.6's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
```

## Linux Commands

1. **ssh:** It is used to securely connect to a device or server over a particular network.

```
nishithat@nishithat-QEMU-Virtual-Machine:~$ ssh nishitha@192.168.64.6
The authenticity of host '192.168.64.6 (192.168.64.6)' can't be established.
ED25519 key fingerprint is SHA256:yuhXsWkC0KyS32Rnt7Mikpv00QQMJF5+hgPW1CbckAs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.64.6' (ED25519) to the list of known hosts.
nishitha@192.168.64.6's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

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.

nishitha@nishitha-QEMU-Virtual-Machine:~$
```

- 
- 2. ssh-keygen:** It is used to generate a set of keys. It has two keys, which are the private key and the public key. The public key can be used to connect to a remote device or server.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nishitha/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nishitha/.ssh/id_rsa
Your public key has been saved in /home/nishitha/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GYDBNyqefmQ6TggjjV7rPlcF+tRdkGldtlSiDl5vrpg nishitha@nishitha-QEMU-Virtual-Machine
The key's randomart image is:
+---[RSA 4096]---+
| ..o. . = o+o |
| o o + +o.. |
| ...+ + + . |
| o .o. * = . |
|= ... .o S . .o |
|+oo = o o |
|.o.B . .
| .*... o . |
| o+=o E . |
+---[SHA256]---+
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

- 3. scp:** (secure copy) It is used to copy files or directories between hosts over SSH protocol.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ scp text.txt nishithat@192.168.64.7:/home/nishithat/text-copy.txt
nishithat@192.168.64.7's password:                                         100%   44    23.3KB/s  00:00
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

```
try: sudo apt install net-tools
nishithat@nishithat-QEMU-Virtual-Machine:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  text-copy.txt  Videos
nishithat@nishithat-QEMU-Virtual-Machine:~$
```

- 4. history:** Gives the list of commands that were previously executed by the user in the current shell session.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ history
1  sudo apt net-tools
2  sudo apt install net-tools
3  sudo apt install ufw
4  sudo ufw status
```

- 
5. **sudo:** It is used to execute commands with root privileges or superuser privileges. It helps to perform administration tasks.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo -u root whoami  
[sudo] password for nishitha:  
root
```

6. **ip:** It is a helpful command that is used to display or configure network interfaces and other features like routing tables and so on. It can also bring up a particular interface.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ip link show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000  
    link/ether 36:00:ed:ed:3a:92 brd ff:ff:ff:ff:ff:ff  
nishitha@nishitha-QEMU-Virtual-Machine:~$ ip addr show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
        inet 127.0.0.1/8 scope host lo  
            valid_lft forever preferred_lft forever  
        inet6 ::1/128 scope host  
            valid_lft forever preferred_lft forever  
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 36:00:ed:ed:3a:92 brd ff:ff:ff:ff:ff:ff  
        inet 192.168.64.6/24 brd 192.168.64.255 scope global dynamic noprefixroute enp0s1  
            valid_lft 83339sec preferred_lft 83339sec  
        inet6 fd01:4bf8:80d1:ad39:a71b:b6dd:da44:a499/64 scope global temporary dynamic  
            valid_lft 601742sec preferred_lft 83019sec  
        inet6 fd01:4bf8:80d1:ad39:93a2:dc76:6e3:2cfe/64 scope global dynamic mngtmpaddr noprefixroute  
            valid_lft 2591977sec preferred_lft 604777sec  
        inet6 fe80::b759:ad24:1f93:be59/64 scope link noprefixroute  
            valid_lft forever preferred_lft forever
```

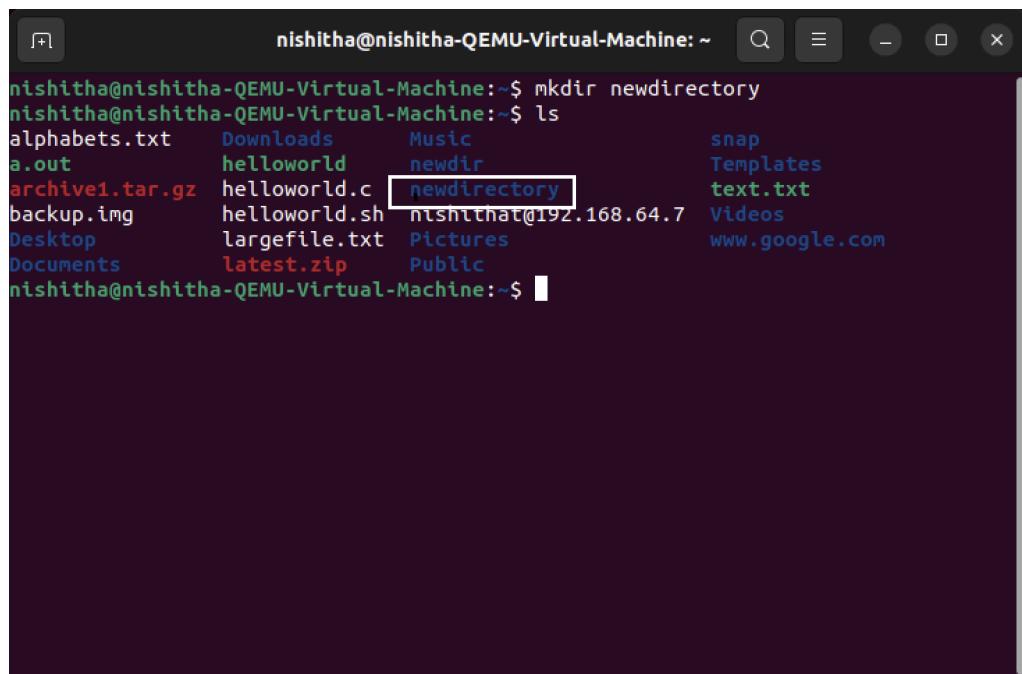
7. **touch:** It is used for creation of files. Multiple file creations are possible using this command. It can also be used to set specific timestamps.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ touch file.txt  
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls  
Desktop  Documents  Downloads  file.txt  Music  nishithat@192.168.64.7  Pictures  Public  snap  Templates  text.txt  Videos  
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

- 8. ls:** It is used to list files and directories. It can list detailed information like hidden files, display file sizes, sort files, grant permissions to file owners, and modify time.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
Desktop Documents Downloads file.txt Music nishithat@192.168.64.7 Pictures Public snap Templates text.txt Videos
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls -l
total 44
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Desktop
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Documents
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Downloads
-rw-rw-r-- 1 nishitha nishitha 0 Jan 17 21:02 file.txt
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Music
-rw-rw-r-- 1 nishitha nishitha 44 Jan 17 20:45 nishithat@192.168.64.7
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Pictures
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Public
drwx----- 3 nishitha nishitha 4096 Jan 17 19:46 snap
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Templates
-rw-rw-r-- 1 nishitha nishitha 44 Jan 17 20:30 text.txt
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Videos
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls -a
. .bash_history .bashrc .config Documents file.txt Music Pictures Public .ssh Templates Videos
.. .bash_logout .cache Desktop Downloads .local nishithat@192.168.64.7 .profile snap .sudo_as_admin_successful text.txt
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

- 9. mkdir:** (make directory) This command is used to create a new directory. It gives the option to create parent and child directories at once.



- 10. cd:** (Change directory) This command is used to go into another directory.



---

**11. dd:** (data duplicator or disk dump) This command is used to convert or copy data and used for creating disk images or copying images between two images.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo dd if=text.txt of=/dev/sda of=backup.img bs=4M
0+1 records in
0+1 records out
44 bytes copied, 0.000191 s, 230 kB/s
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**12. fdisk:** It is used for disk partitioning. It allows users to create, delete, modify and manage partitions on the hard disk.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 69.07 MiB, 72429568 bytes, 141464 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 230.82 MiB, 242032640 bytes, 472720 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 475.09 MiB, 498163712 bytes, 972976 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

**13. apt:** It is a package management tool used for installing, updating, upgrading, and managing software packages.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo apt update
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-proposed InRelease [270 kB]
Hit:3 http://us.ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:4 http://us.ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-proposed/main arm64 Packages [139 kB]
Get:6 http://us.ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [109 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy-proposed/main Translation-en [40.0 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports jammy-proposed/restricted arm64 Packages [149 kB]
Get:9 http://us.ports.ubuntu.com/ubuntu-ports jammy-backports/universe arm64 Packages [22.7 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports jammy-proposed/restricted Translation-en [35.6 kB]
Fetched 994 kB in 1s (782 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**14. vi:** This is a text editor command which is used to manipulate files like create, add, delete or modify files. It is mainly used for system administration or developing codes.

**15. time:** It is used to track the execution time of a particular task or command or a script. It provides information about the real, user and system taken for a particular command.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ time ls -l
total 52
-rw-r--r-- 1 root      root      44 Jan 17 21:13 backup.img
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Desktop
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Documents
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Downloads
-rw-rw-r-- 1 nishitha nishitha     0 Jan 17 21:02 file.txt
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Music
drwxrwxr-x 2 nishitha nishitha 4096 Jan 17 21:03 newdir
-rw-rw-r-- 1 nishitha nishitha    44 Jan 17 20:45 nishithat@192.168.64.7
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Pictures
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Public
drwx----- 3 nishitha nishitha 4096 Jan 17 19:46 snap
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Templates
-rw-rw-r-- 1 nishitha nishitha    44 Jan 17 20:30 text.txt
drwxr-xr-x 2 nishitha nishitha 4096 Jan 17 19:46 Videos

real      0m0.006s
user      0m0.001s
sys       0m0.006s
```

---

**16. tar:** It is used for creating and manipulating archive files. It is used to create compressed files or backing up files in different directories.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ tar -czvf archive1.tar.gz text.txt
text.txt
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
archive1.tar.gz  Desktop   file.txt  nishithat@192.168.64.7  snap      Videos
archive.tar     Documents  Music    Pictures          Templates
backup.img      Downloads  newdir   Public           text.txt
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**17. rm:** (remove) It is used to delete files or directories. It can be used to remove multiple files or directories at once.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
archive1.tar.gz  archive.tar  backup.img  Desktop  Documents  Downloads  file.txt  Music  newdir  nishithat@192.168.64.7  Pictures  Public  snap  Templates  text.txt  Videos
nishitha@nishitha-QEMU-Virtual-Machine:~$ rm archive.tar
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
archive1.tar.gz  backup.img  desktop  Documents  Downloads  file.txt  Music  newdir  nishithat@192.168.64.7  Pictures  Public  snap  Templates  text.txt  Videos
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**18. cat:** (concatenation) It is used for concatenating files and to display the contents of a file. It can also be used to redirect output of files to another file.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
archive1.tar.gz  Documents  Music          Pictures  Templates
backup.img       Downloads  newdir         Public    text.txt
Desktop          file.txt   nishithat@192.168.64.7  snap      Videos
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat text.txt
This is a file created in nishitha machine.
nishitha@nishitha-QEMU-Virtual-Machine:~$ █
```

**19. bash:** The command in Linux is used to run the Bash shell, which is a command processor that provides a command line interface.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat>helloworld.sh
#!/bin/bash
echo "Hello World"
nishitha@nishitha-QEMU-Virtual-Machine:~$ bash helloworld.sh
Hello World
nishitha@nishitha-QEMU-Virtual-Machine:~$ █
```

**20. more:** The command is used to display contents of a text file, one screen at a time. It helps to navigate through the file easily.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls
alphabets.txt      Downloads      Music           snap
a.out              helloworld    newdir          Templates
archive1.tar.gz    helloworld.c newdirectory   text.txt
backup.img         helloworld.sh nishithat@192.168.64.7 Videos
Desktop            largefile.txt Pictures        www.google.com
Documents          latest.zip   Public
nishitha@nishitha-QEMU-Virtual-Machine:~$ more largefile.txt
Hi This is a very large file.
It has lot of words.
It has lot of characters.
This is a file in Linux.
Linux has a kernel and shell.
It is an operating system.
```

**21. watch:** It is used to execute a specific command repeatedly and display the output on a text terminal. It is used for monitoring the changes over time.

```
Every 2.0s: free -m
              0 2024
              total     used     free   shared  buff/cache available
Mem:       3905       726     2324        72       853      2962
Swap:      2629         0     2629

nishitha@nishitha-QEMU-Virtual-Machine: Thu Jan 18 17:12:4
```

**22. ps:** This command gives information about the current processes running in the system. It gives details like the process IDs, resource utilization, etc.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ps
  PID TTY          TIME CMD
 1990 pts/0    00:00:00 bash
 1997 pts/0    00:00:00 ps
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**23. top:** The top command in Linux is a dynamic, real-time system monitoring tool that gives you a live, interactive picture of system resource utilisation. It provides data on processes, system performance, and other system-related metrics.

top - 17:17:13 up 20 min, 1 user, load average: 0.00, 0.03, 0.04													
Tasks: 179 total, 1 running, 178 sleeping, 0 stopped, 0 zombie													
%Cpu(s): 1.0 us, 0.2 sy, 0.0 ni, 98.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st													
MiB Mem : 3905.3 total, 2325.0 free, 725.9 used, 854.4 buff/cache													
MiB Swap: 2630.0 total, 2630.0 free, 0.0 used. 2962.4 avail Mem													
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND		
900	nishitha	20	0	4306664	379556	139208	S	1.7	9.5	0:30.82	gnome-shell		
1	root	20	0	166832	10708	7380	S	0.3	0.3	0:00.48	systemd		
307	systemd+	20	0	15000	6272	5504	S	0.3	0.2	0:02.61	systemd-oomd		
492	root	20	0	15672	7316	6144	S	0.3	0.2	0:00.07	systemd-logind		
1770	root	20	0	0	0	0	I	0.3	0.0	0:00.12	kworker/0:1-events		
1858	nishitha	20	0	513628	55644	41888	S	0.3	1.4	0:00.71	gnome-terminal-		
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthread		
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp		
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp		
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq		
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns		
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:H-events_highpri		
11	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq		
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread		
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread		
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread		
15	root	20	0	0	0	0	S	0.0	0.0	0:00.06	ksoftirqd/0		
16	root	20	0	0	0	0	I	0.0	0.0	0:00.35	rcu_preempt		
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0		
18	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0		
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0		

**24. htop:** The htop command is an interactive process viewer for Unix-like systems that displays system resource utilisation in a user-friendly, colourized format. Htop is a more advanced version of the standard top command.

top - 17:17:13 up 20 min, 1 user, load average: 0.39, 0.16, 0.09													
Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie													
CPU usage (one core): 2.7% user, 3.3% sys, 94.0% idle													
Load average: 0.39 0.16 0.09 Uptime: 00:23:50													
Mem: 806M/3.81G Swap: 0K/2.57G													
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	TIME+	COMMAND			
900	nishitha	20	0	4198M	371M	136M	S	3.3	9.5	0:49.70	/usr/bin/gnome-shell		
936	nishitha	20	0	4198M	371M	136M	S	1.3	9.5	0:14.80	/usr/bin/gnome-shell		
937	nishitha	20	0	4198M	371M	136M	S	1.3	9.5	0:14.69	/usr/bin/gnome-shell		
2588	nishitha	20	0	19932	492	3672	R	1.3	0.1	0:00.28	httpd		
1972	nishitha	20	0	501M	55648	42544	S	0.7	1.4	0:00.94	/libexec/gnome-terminal-server		
1	root	20	0	162M	10708	7380	S	0.0	0.3	0:00.49	/sbin/init splash		
186	root	19	-1	49136	21636	20356	S	0.0	0.5	0:00.27	/lib/systemd/systemd-journald		
228	root	20	0	26632	6272	4096	S	0.0	0.2	0:00.06	/lib/systemd/systemd-udevd		
367	systemd+o	20	0	15000	6272	5504	S	0.0	0.2	0:02.94	/lib/systemd/systemd-oomd		
333	systemd-r	20	0	25224	12672	8448	S	0.0	0.3	0:00.11	/lib/systemd/systemd-resolved		
336	systemd-t	20	0	88740	6272	5504	S	0.0	0.2	0:00.05	/lib/systemd/systemd-timesyncd		
371	systemd-t	20	0	88740	6272	5504	S	0.0	0.2	0:00.00	/lib/systemd/systemd-timesyncd		
467	root	20	0	242M	6892	5996	S	0.0	0.2	0:00.09	/usr/libexec/accounts-daemon		
470	avahl	20	0	7800	3456	3672	S	0.0	0.1	0:00.22	avahl-daemon: running [nishitha-QEMU-Virtual-Machine.local]		
471	root	20	0	17772	2432	2304	S	0.0	0.1	0:00.01	/usr/sbin/cron -f -P		
472	messagebu	20	0	11092	5632	3584	S	0.0	0.1	0:00.23	@ibus-daemon --systemd: --nofork --nopidfile --systemd-activation --syslog-only		
474	root	20	0	264M	17168	14352	S	0.0	0.4	0:00.27	/usr/sbin/NetworkManager --no-daemon		
479	root	20	0	82096	3200	2816	S	0.0	0.1	0:00.16	/usr/sbin/rqbalance --foreground		
481	root	20	0	50316	19846	10624	S	0.0	0.5	0:00.03	/usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers		
482	root	20	0	231M	9164	6272	S	0.0	0.2	0:00.10	/libexec/polkitd --no-debug		
483	root	20	0	242M	6486	5766	S	0.0	0.2	0:00.00	/usr/libexec/power-profiles-daemon		
484	syslog	20	0	216M	4864	3584	S	0.0	0.1	0:00.05	/usr/sbin/syslogd -n -NONE		
487	root	20	0	1287M	28724	18176	S	0.0	0.7	0:00.83	/usr/lib/snapd/snapd		
488	root	20	0	239M	5766	5248	S	0.0	0.1	0:00.00	/libexec/switcheroo-control		
489	root	20	0	82096	3200	2816	S	0.0	0.1	0:00.00	/usr/sbin/rqbalance --foreground		
490	root	20	0	231M	9164	6272	S	0.0	0.2	0:00.00	/libexec/polkitd --no-debug		
492	root	20	0	15672	7316	6144	S	0.0	0.2	0:00.07	/lib/systemd/systemd-logind		
493	root	20	0	384M	11732	9428	S	0.0	0.3	0:00.05	/libexec/wdisks2/udisks2		
500	root	20	0	15244	5632	4864	S	0.0	0.1	0:00.02	/bin/wpa_supplicant -u -s -0 /run/wpa_supplicant		
564	root	20	0	239M	5766	5248	S	0.0	0.1	0:00.00	/libexec/switcheroo-control		

**25. gcc:** The gcc command compiles programming languages such as C and C++. It is a component of the GNU Compiler Collection and is frequently used to convert source code into executable programmes.

```
nisitha@nishitha-QEMU-Virtual-Machine:~$ cat>helloworld.c
#include<stdio.h>
int main()
{
    printf("Hello World");
    return 0;
}
nisitha@nishitha-QEMU-Virtual-Machine:~$ gcc helloworld.c
nisitha@nishitha-QEMU-Virtual-Machine:~$ ./a.out
Hello World
nisitha@nishitha-QEMU-Virtual-Machine:~$
```

**26. tail:** The tail command displays the last few lines of a text file. It is particularly useful for reading log files and tracking changes in real time.

```
nisitha@nishitha-QEMU-Virtual-Machine:~$ cat>largefile.txt
Hi This is a very large file.
It has lot of words.
It has lot of characters.
This is a file in Linux.
Linux has a kernel and shell.
It is an operating system.
nisitha@nishitha-QEMU-Virtual-Machine:~$ tail -2 largefile.txt
Linux has a kernel and shell.
It is an operating system.
nisitha@nishitha-QEMU-Virtual-Machine:~$
```

**27. grep:** The grep command is a strong tool for searching and pattern matching in text files. It is often used alongside other commands or within scripts.

```
nisitha@nishitha-QEMU-Virtual-Machine:~$ grep is largefile.txt
Hi This is a very large file.
This is a file in Linux.
It is an operating system.
```

**28. kill:** The kill command terminates or signals a process. It provides a signal to a specific process or group of processes, giving you control over their behaviour.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE      14) SIGALRM     15) SIGTERM
16) SIGSTKFLT    17) SIGCHLD     18) SIGCONT      19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG       24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1   36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6   41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11  46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14  51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9   56) SIGRTMAX-8   57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4   61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

**29. killall:** The killall command terminates or signals processes using their names rather than their process IDs. It allows you to easily terminate numerous processes with the same name.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ killall -l
HUP INT QUIT ILL TRAP ABRT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STI
CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF WINCH POLL PWR SYN
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

**30. du:** The du command estimates the amount of space utilised by files and directories. It displays the disk space used by files and folders, either individually or in a summary format.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ du ~
4          /home/nishitha/Documents
4          /home/nishitha/Templates
4          /home/nishitha/.local/share/sounds
12         /home/nishitha/.local/share/keyrings
8          /home/nishitha/.local/share/gnome-shell
76         /home/nishitha/.local/share/gvfs-metadata
4          /home/nishitha/.local/share/evolution/calendar/system
4          /home/nishitha/.local/share/evolution/calendar/trash
12         /home/nishitha/.local/share/evolution/calendar
4          /home/nishitha/.local/share/evolution/memos/trash
8          /home/nishitha/.local/share/evolution/memos
4          /home/nishitha/.local/share/evolution/addressbook/system/photos
92         /home/nishitha/.local/share/evolution/addressbook/system
4          /home/nishitha/.local/share/evolution/addressbook/trash
100        /home/nishitha/.local/share/evolution/addressbook
8          /home/nishitha/.local/share/evolution/tasks/system
4          /home/nishitha/.local/share/evolution/tasks/trash
16         /home/nishitha/.local/share/evolution/tasks
4          /home/nishitha/.local/share/evolution/mail/trash
8          /home/nishitha/.local/share/evolution/mail
148        /home/nishitha/.local/share/evolution
4          /home/nishitha/.local/share/flatpak/db
8          /home/nishitha/.local/share/flatpak
4          /home/nishitha/.local/share/ibus-table
4          /home/nishitha/.local/share/gnome-settings-daemon
12         /home/nishitha/.local/share/icc
```

---

**31. df:** The df command displays information on disk space usage on file systems. It displays the total disk space, utilised space, available space, and percentage of used space on mounted file systems. The -h option displays disk space consumption in a human-readable way.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          391M   1.7M  389M   1% /run
/dev/vda2        24G   11G   13G  45% /
tmpfs          2.0G     0  2.0G   0% /dev/shm
tmpfs          5.0M   4.0K  5.0M   1% /run/lock
efivarfs       256K   27K  230K  11% /sys/firmware/efi/efivars
/dev/vda1        512M   5.3M  507M   2% /boot/efi
tmpfs          391M   92K  391M   1% /run/user/1000
```

**32. screen:** The screen command is a terminal multiplexer that allows you to run numerous terminal sessions from the same terminal window or remote terminal. It supports session persistence, detachable sessions, and session sharing.

```
GNU Screen version 4.09.00 (GNU) 30-Jan-22
Copyright (c) 2018-2020 Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2010-2014 Juergen Weigert, Sadrul Habib Chowdhury
Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan, Sadrul Habib Chowdhury
Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program (see the file COPYING); if not, see https://www.gnu.org/licenses/, or contact Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02111-1301 USA.

Send bugreports, fixes, enhancements, t-shirts, money, beer & pizza to screen-devel@gnu.org

Capabilities:
+copy +remote-detach +power-detach +multi-attach +multi-user +font +color-256 +utf8 +rxvt +builtin-telnet
```

**33. vim:** The vim command launches the Vim text editor, which is highly configurable and powerful. Vim is renowned for its speed, adaptability, and comprehensive feature set.

**34. chmod:** The chmod command changes the permissions of a file or directory.

It gives you control over who can read, write, and execute a file or directory.

To set permissions, use the `chmod` command, which takes symbolic or octal values.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls -l text.txt  
-rw-rw-r-- 1 nishitha nishitha 44 Jan 17 20:30 text.txt  
nishitha@nishitha-QEMU-Virtual-Machine:~$ chmod 755 text.txt  
nishitha@nishitha-QEMU-Virtual-Machine:~$ ls -l text.txt  
-rwxr-xr-x 1 nishitha nishitha 44 Jan 17 20:30 text.txt  
nishitha@nishitha-QEMU-Virtual-Machine:~$ █
```

**35. chown:** The chown command changes the ownership of files and directories.

It enables you to modify the user and/or group ownership of one or more files or directories.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ chown root text.txt  
chown: changing ownership of 'text.txt': Operation not permitted
```

---

**36. useradd:** The useradd command creates a new user account. It is commonly used by system administrators to add users to the system.

```
nishitha@nishitha-QEMU-Virtual-Machine: $ sudo useradd vijay
[sudo] password for nishitha:
nishitha@nishitha-QEMU-Virtual-Machine: $ cut -d: -f1 /etc/passwd | sort
_apt
avahi
avahi-autoipd
backup
bin
colorl
cups-pk-helper
daemon
dnsmasq
fwupd-refresh
games
gdm
geoclue
gnats
gnome-initial-setup
hplip
irc
kernoops
llist
lp
mail
man
messagebus
news
nishitha
nm-openvpn
nobody
proxy
pulse
root
rtkit
saned
speech-dispatcher
sshd
sssd
sync
sys
syslog
systemd-network
systemd-oom
systemd-resolve
systemd-timesync
tcpdump
tts
usbdux
upnp
utld
vijay > It is here
whoopsie
www-data
```

**37. mv:** The mv command moves or renames files and folders. It allows you to move and rename files and directories.

```
nishitha@nishitha-QEMU-Virtual-Machine: $ ls
a.out      backup.img  Documents  file.txt  helloworld.c  largefile.txt  newdir      Pictures  snap      text.txt
archive1.tar.gz  Desktop   Downloads  helloworld  helloworld.sh  Music       nishithat@192.168.64.7  Public    Templates  Videos
nishitha@nishitha-QEMU-Virtual-Machine: $ mv file.txt newdir
nishitha@nishitha-QEMU-Virtual-Machine: $ ls'
> ;
> '
ls
;
: command not found
nishitha@nishitha-QEMU-Virtual-Machine: $ ls
a.out      backup.img  Documents  helloworld  helloworld.sh  Music       nishithat@192.168.64.7  Public    Templates  Videos
archive1.tar.gz  Desktop   Downloads  helloworld.c  largefile.txt  newdir      Pictures      snap      text.txt
nishitha@nishitha-QEMU-Virtual-Machine: $ ls newdir
file.txt
```

**38. man:** The man command displays instructional pages for other commands and system files. Manual pages, often known as man pages, include thorough documentation and information on many commands, utilities, and system calls.

The screenshot shows a terminal window with the title "nishitha@nishitha-QEMU-Virtual-Machine: ~". The window displays the man(1) manual page for the "man" command. The page includes sections for NAME, SYNOPSIS, DESCRIPTION, EXAMPLES, and NOTES. It lists various options for the man command, such as -k for apropos, -K for man2man, -f for whatis, and -l for less. The SYNOPSIS section shows examples like "man [man options] [[section] page ...]". The DESCRIPTION section explains that man is the system's manual pager and provides a table of section numbers and their types. The EXAMPLES section shows "man ls" and "Display the manual page for the item (program) ls.". The NOTES section contains conventions for bold, italic, and code text.

```

MAN(1)                                         Manual pager utils                                         MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...]
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

    The table below shows the section numbers of the manual followed by the types of pages they contain.

1   Executable programs or shell commands
2   System calls (functions provided by the kernel)
3   Library calls (functions within program libraries)
4   Special files (usually found in /dev)
5   File formats and conventions, e.g. /etc/passwd
6   Games
7   Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7), man-pages(7)
8   System administration commands (usually only for root)
9   Kernel routines [Non standard]

A manual page1 consists of several sections.

Conventional section names include NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUE, ERRORS, ENVIRONMENT, FILES, VERSIONS, CONFORMING TO, NOTES, BUGS, EXAMPLE, AUTHORS, and SEE ALSO.

The following conventions apply to the SYNOPSIS section and can be used as a guide in other sections.

bold text          type exactly as shown.
italic text        replace with appropriate argument.
[abc]           any or all arguments within [ ] are optional.
-a|-b            options delimited by | cannot be used together.
argument ...     argument is repeatable.
[expression] ... entire expression within [ ] is repeatable.

Exact rendering may vary depending on the output device. For instance, man will usually not be able to render italics when running in a terminal, and will typically use underlined or coloured text instead.

The command or function illustration is a pattern that should match all possible invocations. In some cases it is advisable to illustrate several exclusive invocations as is shown in the SYNOPSIS section of this manual page.

EXAMPLES
    man ls
        Display the manual page for the item (program) ls.

Man page man(1) line 1 (press h for help or q to quit).

```

**39. locate:** The locate command is used to quickly determine the location of files and directories on the file system. It uses a pre-built index of the system's files, making it faster than other search methods.

The screenshot shows a terminal window with the title "nishitha@nishitha-QEMU-Virtual-Machine: ~\$". The command "locate file.txt" is run, and the output shows three file paths: "/home/nishitha/largefile.txt", "/home/nishitha/newdir/file.txt", and "/usr/share/doc/alsa-base/driver/Procfile.txt.gz".

```

nishitha@nishitha-QEMU-Virtual-Machine:~$ locate file.txt
/home/nishitha/largefile.txt
/home/nishitha/newdir/file.txt
/usr/share/doc/alsa-base/driver/Procfile.txt.gz

```

**40. find:** The find command is a flexible tool for finding and locating files and directories in a directory hierarchy using a variety of parameters.

The screenshot shows a terminal window with the title "nishitha@nishitha-QEMU-Virtual-Machine:~\$". The command "find . -name text.txt" is run, and the output shows a single file path: "./text.txt".

```

nishitha@nishitha-QEMU-Virtual-Machine:~$ find . -name text.txt
./text.txt

```

**41. sed:** The sed command is a stream editor that may do simple text modifications on an input stream. It is an effective and adaptable tool for text processing and manipulation.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat text.txt
This is a file created in nishitha machine.
nishitha@nishitha-QEMU-Virtual-Machine:~$ sed '1i\This is a new line added by sed.' text.txt
This is a new line added by sed.
This is a file created in nishitha machine.
```

**42. awk:** The awk command is an advanced text processing tool used to scan and parse patterns. It is especially useful for processing and analysing text data in files or from command output.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ awk 'END{print NR}' text.txt
1
```

**43. diff:** The diff command compares and identifies differences between two files or folders. It is a useful tool for detecting changes in text files or directories.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ diff text.txt largefile.txt
1c1,6
< This is a file created in nishitha machine.
---
> Hi This is a very large file.
> It has lot of words.
> It has lot of characters.
> This is a file in Linux.
> Linux has a kernel and shell.
> It is an operating system.
```

**44. sort:** The sort command sorts text files or pipeline input. It allows you to sort files or text streams in either ascending or descending order.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat>alphabets.txt
q
f
a
y
t
u
d
n
x
c
j
nishitha@nishitha-QEMU-Virtual-Machine:~$ sort alphabets.txt
a
c
d
f
j
k
n
q
t
u
y
```

---

**45. export:** The export command in Linux is used to define an environment variable. Environment variables are variables that can be used to customise the behaviour or configuration of shell processes and programmes.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ export MY_VARIABLE="Hello World"
```

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ env  
SHELL=/bin/bash  
MY_VARIABLE=Hello World  
QT_IM_MODULE=ibus  
XDG_RUNTIME_DIR=/run/user/1000
```

**46. pwd:** The pwd command in Linux prints the current working directory. The pwd command displays the entire path to the directory you are currently in.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ pwd  
/home/nishitha
```

**47. crontab:** The crontab command is used to set up, change, and administer cron jobs. Cron is a time-based job scheduler in Unix-like operating systems that allows users to schedule jobs or commands to execute at specific times, dates, or intervals.

```
GNU nano 6.2          /tmp/crontab.lpsrdw/crontab  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# Read 23 lines ]  
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location  
^X Exit      ^R Read File  ^L Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

---

**48. mount:** The mount command is used to associate a file system or storage device with a specified directory in the file system hierarchy. This is an essential command for mounting a variety of file systems, including those on external drives, network shares, and special device files.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1943504k,nr_inodes=485876,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=020,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=399900k,mode=755,inode64)
/dev/vda2 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=16916)
hugepages on /dev/hugepages type hugepages (rw,relatime,page size=2M)
mqqueue on /dev/mqueue type mqqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
ramfs on /run/credentials/systemd-sysusers.service type ramfs (ro,nosuid,nodev,noexec,relatime,mode=700)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
```

**49. passwd:** The passwd command changes the password of a user account. It allows users to set and change their passwords interactively.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ passwd
Changing password for nishitha.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

**50. uname:** The uname command displays system information. It includes information on the operating system's name, architecture, kernel version, and other system-related facts.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ uname
Linux
```

**51. whereis:** The whereis command locates the binary, source, and manual page files for a command. It allows you to locate the many components connected with a specific programme or command.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ whereis -s ls
ls:
```

---

**52. whatis:** The whatis command provides a brief description or summary of a given command. It summarises the purpose and capabilities of the given command.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ whatis ls
ls (1)           - list directory contents
```

**53. less:** The less command is a pager tool that displays the contents of text files in a scrollable format. It enables users to traverse files, search for specific text, and execute various operations on displayed text.

```
1 Hi This is a very large file.
2 It has lot of words.
3 It has lot of characters.
4 This is a file in Linux.
5 Linux has a kernel and shell.
6 It is an operating system.

[...]
```

I

**54. su:** After login, use the su command to switch to another user account, generally superuser or root. It allows a user to execute commands with the privileges of another user.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo su root
[sudo] password for nishitha:
Sorry, try again.
[sudo] password for nishitha:
root@nishitha-QEMU-Virtual-Machine:/home/nishitha#
```

---

**55. ping:** The ping command is used to determine whether a host (often a computer or server) is reachable via an Internet Protocol (IP) network. It also calculates the round-trip time of communications delivered from the originating host to a destination computer and back.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.208 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.417 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.162 ms
```

**56. traceroute:** The traceroute command is used to track the path packets follow to reach a destination host on an IP network. It offers information about the intermediate routers, or hops, that connect the source and destination.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ sudo traceroute google.com
traceroute to google.com (172.217.4.78), 30 hops max, 60 byte packets
1 _gateway (192.168.64.1) 0.314 ms 0.278 ms 0.260 ms
2 puma7-atom.ht.home (192.168.0.1) 4.905 ms 5.428 ms 5.392 ms
3 10.20.0.1 (10.20.0.1) 20.112 ms 30.261 ms 23.719 ms
4 216.80.78.91 (216.80.78.91) 30.129 ms 30.091 ms 30.057 ms
5 207.172.17.3 (207.172.17.3) 30.011 ms hge0-0-0-15.core1.chgo.il.rcn.net (207.172.18.38) 29.990 ms 207.172.17.3 (207.172.17.3) 29.958 ms
6 hge0-0-0-0.border1.eqnx.il.rcn.net (207.172.19.163) 29.944 ms 21.011 ms hge0-0-0-4.border1.eqnx.il.rcn.net (207.172.19.171) 20.988 ms
7 72.14.205.48 (72.14.205.48) 24.154 ms 19.769 ms 19.753 ms
8 * * *
9 142.251.61.40 (142.251.61.40) 24.420 ms 142.251.60.202 (142.251.60.202) 24.413 ms 108.170.243.174 (108.170.243.174) 24.402 ms
10 ord37s18-in-f14.1e100.net (172.217.4.78) 24.376 ms 108.170.243.219 (108.170.243.219) 24.369 ms 216.239.51.117 (216.239.51.117) 24.363 ms
```

**57. date:** The date command in Linux displays or sets the system date and time. It gives a simple way to view the current date and time, as well as change the system clock if necessary.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ date
Sat Jan 20 02:26:22 PM CST 2024
```

**58. wget:** The wget command is a tool for getting files from the Internet. It supports a variety of protocols, including HTTP, HTTPS, and FTP, allowing you to download files from remote servers.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ wget -r www.google.com
] -2024-01-18 20:26:48-- http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.190.4, 2607:f8b0:4009:803::2004
Connecting to www.google.com (www.google.com)|142.250.190.4|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'www.google.com/index.html'

www.google.com/index.html [ <=> ] 17.84K ---KB/s in 0.03s

2024-01-18 20:26:48 (652 KB/s) - 'www.google.com/index.html' saved [18272]

Loading robots.txt; please ignore errors.]
--2024-01-18 20:26:48-- http://www.google.com/robots.txt
Reusing existing connection to www.google.com:80.
HTTP request sent, awaiting response... 200 OK
Length: 8371 (8.2K) [text/plain]
Saving to: 'www.google.com/robots.txt'

www.google.com/robots.txt [ ==> ] 8.17K ---KB/s in 0.008s

2024-01-18 20:26:48 (1.03 MB/s) - 'www.google.com/robots.txt' saved [8371/8371]
```

**59. wc:** The wc command counts the number of lines, words, and bytes in a file or standard input. It returns basic statistics on the contents of a text file.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ wc largefile.txt
6 34 159 largefile.txt
```

**60. clear:** The clear command clears the terminal screen by erasing all previous commands and outputs. It clears the terminal window, allowing fresh commands to be entered.

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ clear
```

```
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

## BASH SCRIPTS:

1. Write a script called “disk-benchmark-background.sh” that uses the dd command to run a benchmark against the local disk in the background, that captures all the output (both standard out and error output) to a file “disk-benchmark-background-log.txt”. Use the “time” command to show how long the benchmark took to complete. The benchmark should run for at least 10 seconds, and it should complete even if the ssh (or bash) session is terminated.

```

nishitha@nishitha-QEMU-Virtual-Machine:~$ cat disk_benchmark_background.sh
#!/bin/bash
echo "Running the dd command in the background..."
dd if=/dev/zero of=output.txt bs=1M count=5000 conv=fdatasync > disk_benchmark_log.txt 2>&1 &
echo "The dd command is running in the background."
sleep 10
nishitha@nishitha-QEMU-Virtual-Machine:~$ time bash disk_benchmark_background.sh
Running the dd command in the background...
The dd command is running in the background.

real    0m10.020s
user    0m0.014s
sys     0m1.899s
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat disk_benchmark_log.txt
5000+0 records in
5000+0 records out
5242880000 bytes (5.2 GB, 4.9 GiB) copied, 1.97853 s, 2.6 GB/s
nishitha@nishitha-QEMU-Virtual-Machine:~$
```

2. Write a script called “network-test.sh” that takes input a file “network-test-machinelist.txt” with a list of DNS names (e.g. google.com, iit.edu, anl.gov), each name on a separate line, and runs the ping utility collecting 3 samples from each DNS name, and writing the RTT (round trip time) average latency into a file “network-test-latency.txt” where each line will have the DNS name and average RTT separated by a space. Make sure it works with at least 10 DNS names, but it should work for an unspecified number of DNS names.

```

nishitha@nishitha-QEMU-Virtual-Machine:~$ cat>network-test-machinelist.txt
google.com
wikipedia.org
iit.edu
transitchicago.com
youtube.com
gmail.com
linux.com
w3schools.com
leetcode.com
127.0.0.1

nishitha@nishitha-QEMU-Virtual-Machine:~$ cat network-test.sh
#!/bin/bash
if [ ! -f "network-test-machinelist.txt" ]; then
    echo "Error: Input file network-test-machinelist.txt not found."
    exit 1
fi

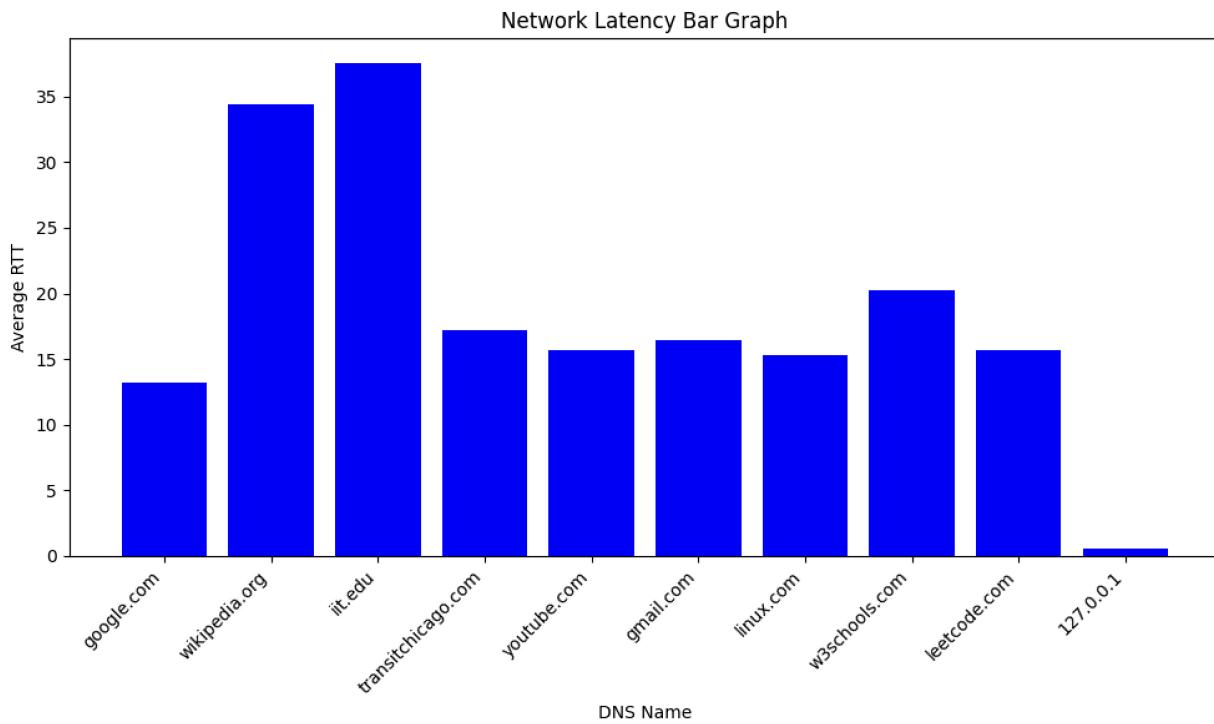
while IFS= read -r dns_name; do
    if [ -n "$dns_name" ]; then
        avg_rtt=$(ping -c 3 "$dns_name" | tail -1 | awk '{print $4}' | cut -d '/' -f 2)
        echo "$dns_name $avg_rtt" >> "network-test-latency.txt"
    fi
done < "network-test-machinelist.txt"
echo "Network test completed.."
```

```
nishitha@nishitha-QEMU-Virtual-Machine:~$ bash network-test.sh
Network test completed..
nishitha@nishitha-QEMU-Virtual-Machine:~$ cat network-test-latency.txt
google.com 13.211
wikipedia.org 34.418
iit.edu 37.532
transitchicago.com 17.155
youtube.com 15.635
gmail.com 16.465
linux.com 15.344
w3schools.com 20.256
leetcode.com 15.721
127.0.0.1 0.528
```

3. Write a Python matplotlib script to generate a graph of the "network-test-latency.txt" data. The graph should automatically adjust to the number of entries, and the scale of the data.

```
import matplotlib.pyplot as plt
dns_names = []
avg_rtt_values = []
with open('network-test-latency.txt', 'r') as file:
    lines = file.readlines()
for line in lines:
    name,value = line.split()
    dns_names.append(name)
    avg_rtt_values.append(float(value))

plt.figure(figsize=(10, 6))
plt.bar(dns_names, avg_rtt_values, color='blue')
plt.xlabel('DNS Name')
plt.ylabel('Average RTT')
plt.title('Network Latency Bar Graph')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



## Theory Questions

1. In the system configuration of the VM, explain how changing the number of processors changes the behavior of your VM. Explain a scenario where you want to set this to the minimum, and a scenario where you want to set it to the maximum. Why is setting it to the maximum potentially a bad idea?

### Ans:

- Changing the number of processors.
  - i. Scenario of Maximum Processors:
    - Use Case: For resource-intensive apps or workloads that gain advantages from parallel processing.
    - Impact: Improved performance and faster completion of parallelizable jobs.
    - Examples include running big databases, scientific simulations, and rendering programmes.
    - Considerations:

- 
- Increased resource utilisation by the host system.
    - Competing for CPU resources with other VMs on the identical host.
  - ii. Scenario for the minimum processors:
    - Use Case: For lightweight programmes or operations that do not require several CPUs.
    - Impact: Reduced resource use, which freed up resources for other virtual machines.
    - Examples include Web servers, basic apps, and low-demand services.
    - Considerations:
      - Limited parallel processing capacity.
      - Possible underuse of available resources.
  - Possible Issues with Setting to Maximum:
    - i. Resource Contention: Allocating an excessive number of processors to a VM may cause resource congestion on the host system. Other VMs on the very same host may suffer performance deterioration owing to competition for CPU resources.
    - ii. Overhead: Each vCPU allocated to a virtual machine incurs some expense. Setting the number of processors to their maximum may result in overhead that outweighs the speed savings, particularly if the workload is not highly parallelizable.
    - iii. Scaling Challenges: The rule of decreasing returns frequently applies when increasing the total number of processors. After a certain extent, adding additional processors might not end up resulting in a proportional improvement in performance. In other circumstances, it may even result in lower returns or more complexity.

- 
2. In the system configuration of the VM, under the Acceleration Tab, explain the difference between the paravirtualization options: None, Legacy, Minimal, Hyper-V, and KVM. Explain which one would be best to use with Ubuntu Linux and why.

**Ans:**

- a. None:
  - i. This option indicates that no paravirtualization is enabled. The VM operates purely in software emulation mode.
  - ii. Performance may be lower than with paravirtualized choices.
- b. Legacy:
  - i. This often implies legacy virtualization techniques, which frequently lack particular optimisations for the latest hardware and virtualization extensions.
  - ii. Performance may be inferior when compared with newer paravirtualization options.
- c. Minimal:
  - i. Minimal paravirtualization refers to a lightweight or basic collection of optimisations for improving virtualization speed.
  - ii. It may not have all of the functionality available in more powerful paravirtualization alternatives.
- d. Hyper-V:
  - i. Hyper-V is a virtualization platform created by Microsoft.
  - ii. Hyper-V paravirtualization may contain optimisations tailored for integration with Microsoft Hyper-V.
- e. Kernel-based Virtual Machine (KVM):
  - i. KVM is a Linux kernel module that enables hardware-assisted virtualization.
  - ii. KVM enables the VM to take benefit of the hardware virtualization extensions on the host system, resulting in increased performance.

- 
- The best option for Ubuntu Linux guests is to use KVM as the paravirtualization option. Here's why:
    - Native Support: KVM is a component of the Linux kernel that uses hardware virtualization extensions to achieve peak performance. This yields efficient and native support for virtualization on Linux hosts.
    - Performance: KVM generally provides outstanding performance for Linux guests because it enables the VM to directly use hardware virtualization features.
    - Integration: Ubuntu is a Linux distribution, and KVM is seamlessly integrated with the Linux kernel. This integration improves compatibility and provides smooth operation for Ubuntu Linux guests.
3. In storage devices when configuring the VM, there are multiple types of storage controllers: explain the difference between the IDE, SATA, and NVMe controller. Give an example for each type of storage controller of a scenario where you may want to use this type of controller.

**Ans:**

- a. IDE (Integrated Drive Electronics): Specifications:
  - i. Previously used to connect hard discs and optical drives in personal computers.
  - ii. Supports a maximum of two devices per channel (primary and secondary).
  - iii. Data transfer rates may be limited when compared to newer controllers.
  - iv. Example scenario:
    - Scenario: Running a virtual machine with outdated operating systems or apps that provide support for IDE devices.
    - Use Case: Maintaining an outdated operating system with no drivers for contemporary storage controllers.

- 
- b. SATA (Serial ATA): Specifications:
    - i. IDE was replaced as the preferred method for interfacing storage devices.
    - ii. Supports larger data transfer rates than IDE.
    - iii. In personal computers, HDDs and SSDs are commonly utilized.
    - iv. Example scenario:
      - Scenario: Setting up a virtual machine (VM) for everyday computing or file storage.
      - Use Case: Running a virtual machine that demands an appropriate level of performance and compatibility without the necessity for NVMe's ultra-high speeds.
  - c. NVMe (Non-Volatile Memory Express): Specifications:
    - i. Designed for flash-based storage devices, it provides much greater data transfer rates than SATA.
    - ii. Optimized for low-latency, high-throughput access to non-volatile memory (such as SSDs).
    - iii. Typically applied to high-performance computing and data-intensive tasks.
    - iv. Example scenario:
      - Scenario: Running a VM with demanding tasks that require fast storage access.
      - Databases, data analytics, or virtual desktop infrastructure (VDI) require quick storage access for optimal performance.
4. In the network configuration of the VM, there are multiple types of network adapters: explain the difference between NAT, Bridged Adapter, Internal Network, and Host-only Network. Give an example for each type of network of a scenario where you may want to use this type of network.

**Ans:**

- 
- a. NAT (Network Address Translation): Characteristics:
    - i. The VM has the same IP address as the host and appears to be connected to the same network.
    - ii. Outbound internet access is given via the host's network connection.
    - iii. Inbound connections are usually not readily available from external networks.
    - iv. Example scenario:
      - Scenario: Running a virtual machine that requires internet access but not direct external network exposure.
      - Use Case: Testing software upgrades or web browsing within the virtual machine.
  - b. Bridged Adapter: Characteristics:
    - i. The VM is assigned its own IP address on the same network as the host.
    - ii. On the network, the virtual machine appears as a separate device.
    - iii. Allows direct communication between the VM and other devices on the exact same network.
    - iv. Example scenario:
      - Scenario: Running a server virtual machine (VM) that must be directly accessible from other devices on the local network.
      - Use Case: Hosting a web server, a file server, or any other service that requires remote access.
  - c. Internal Network: Characteristics:
    - i. VMs on the internal network can communicate with one another, but not with the host or external networks.
    - ii. Isolated network for communication among VMs running on the same hypervisor.
    - iii. Example scenario:
      - Scenario: Creating a test environment in which VMs must communicate with one another but do not have external connectivity.

- 
- Use Case: Running a multi-tier application with distinct virtual machines for the front end, back end, and database.
  - d. Host-Only Network: Characteristics:
    - i. VMs on a host-only network can connect with one another and the host, but not with other networks.
    - ii. Provides network separation for virtual machines from external networks.
    - iii. Example scenario:
      - Scenario: Creating a development environment in which virtual machines (VMs) communicate with one another and with the host but do not require external connectivity.
      - Use Case: Create and test a networked application that does not require internet access during development.
  - 5. For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers.

**Ans:**

- a. USB 1.1 Controller:
  - i. Data Transfer Speed:
    - Maximum data transfer rate is 12 Mbps.
  - ii. Characteristics:
    - This is the first generation USB technology.
  - iii. Typically used to connect simple peripherals such as keyboards, mouse, and printers.
  - iv. Because of its restricted bandwidth, this device is best suited for low-speed applications.
- b. USB 2.0 Controller:
  - i. Data Transfer Speed:
    - Maximum rate of 480 Mbps.
  - ii. Characteristics:
    - Compared to USB 1.1, this version offers faster data transfers.
    - External hard discs, cameras, and USB flash drives are all commonly used accessories.

- 
- Backward compatible with USB 1.1 devices (connects at USB 1.1 speeds).
  - c. USB 3.0 controller:
    - i. Data Transfer Speed:
      - USB 3.0, commonly known as USB 3.1 Gen 1, can carry data at up to 5 Gbps.
    - ii. Characteristics:
      - Significant increase in data transfer speeds over USB 2.0.
      - Ideal for high-speed data transfer applications like external hard drives and high-definition video cameras.
      - Backward compatible with USB 2.0 and USB 1.1 devices (connects at reduced speeds).

## References

1. <https://community.sophos.com/intercept-x-endpoint/f/recommended-reads/142324/linux-command-reference-guide>
2. <https://github.com/The-Order-of-Penguins/Books/blob/master/linux-command-line-and-shell-scripting-bible-by-richard-blum-christine-bresnahan.pdf>
3. <https://www.freecodecamp.org/news/the-linux-commands-handbook/>
4. <https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/>