

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. To the right of the title is a photograph of a server room with multiple server racks. The main content area contains the following text:

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.



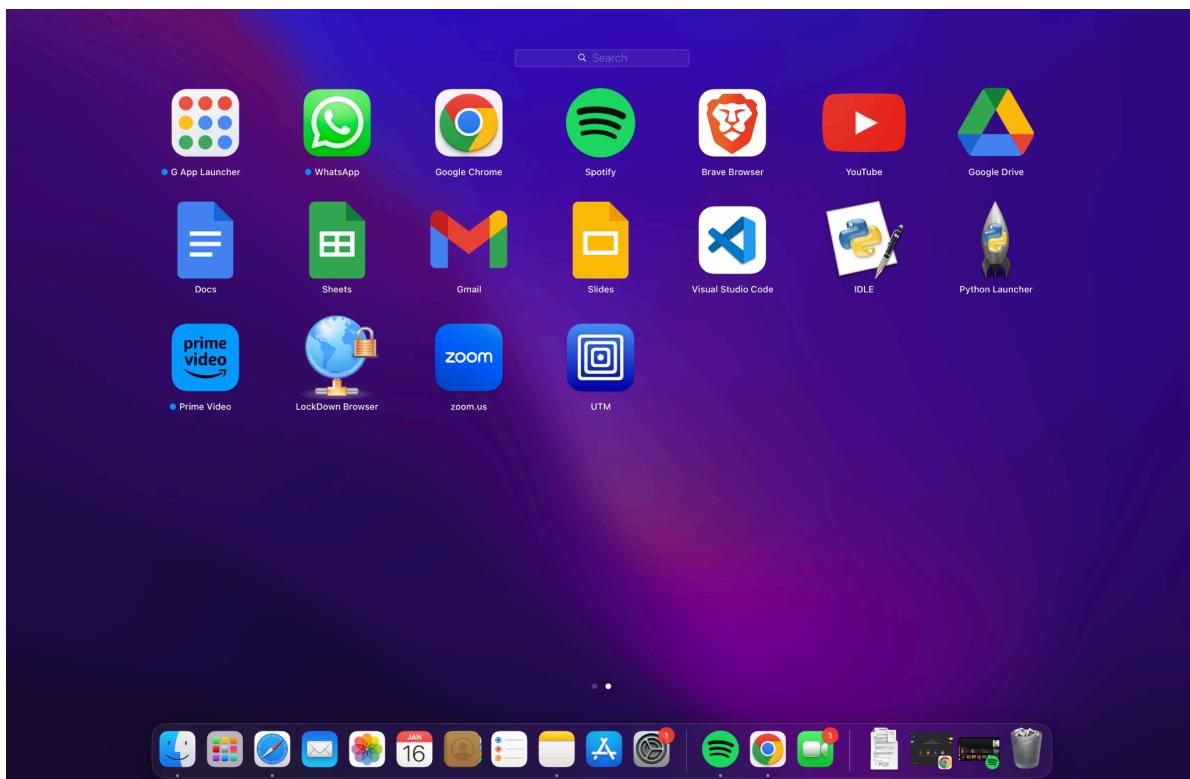
ORACLE

Oracle VM VirtualBox Overview

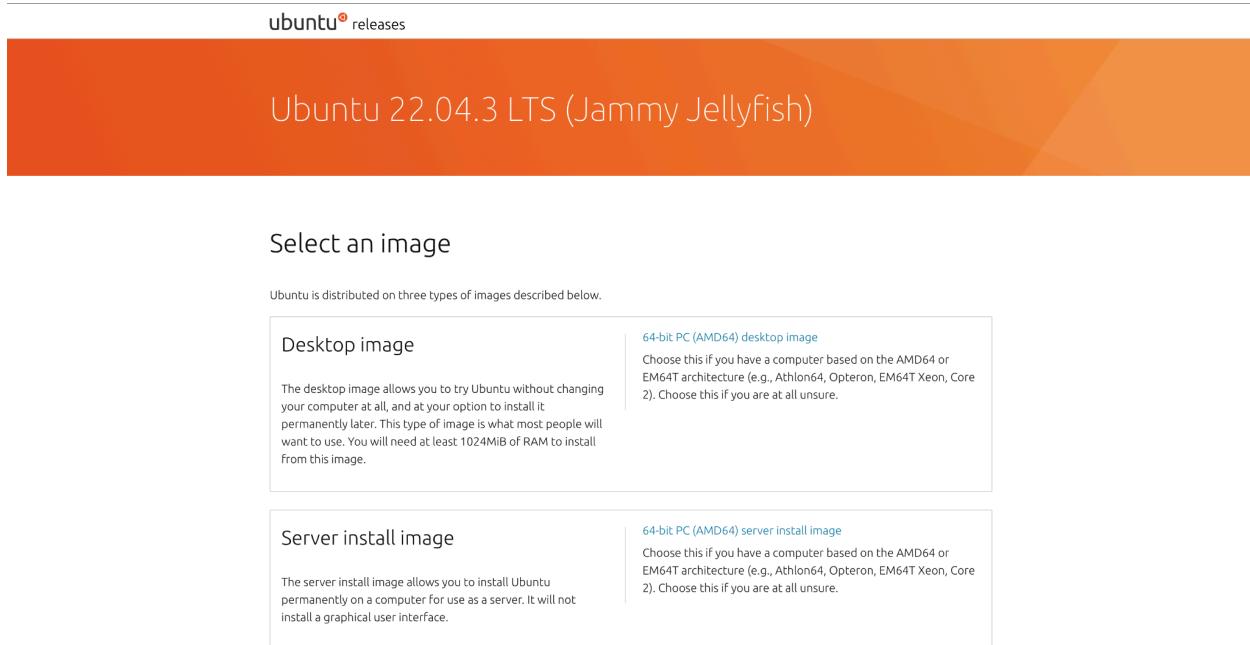
An Oracle White Paper

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

2. Download UTM

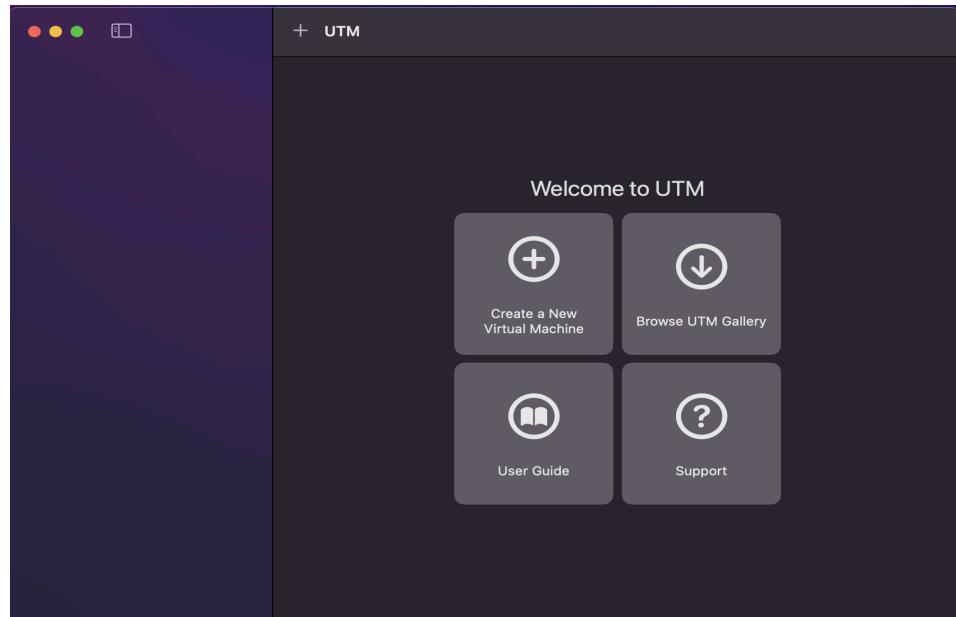


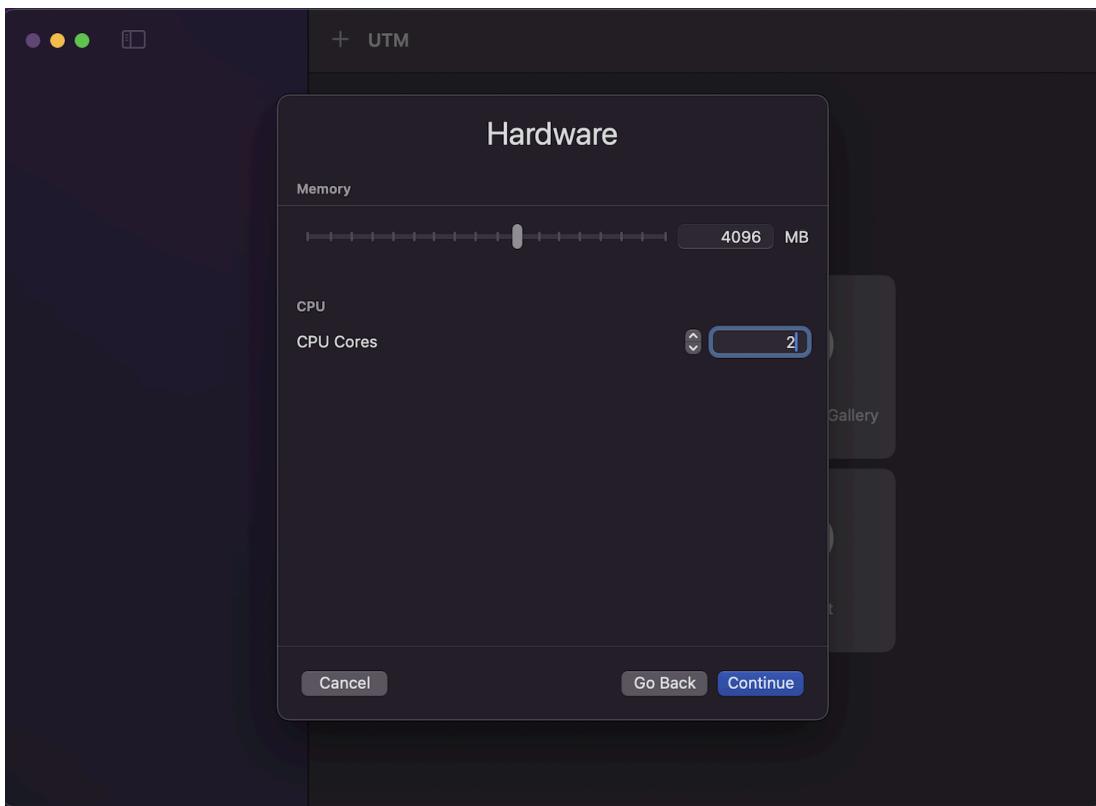
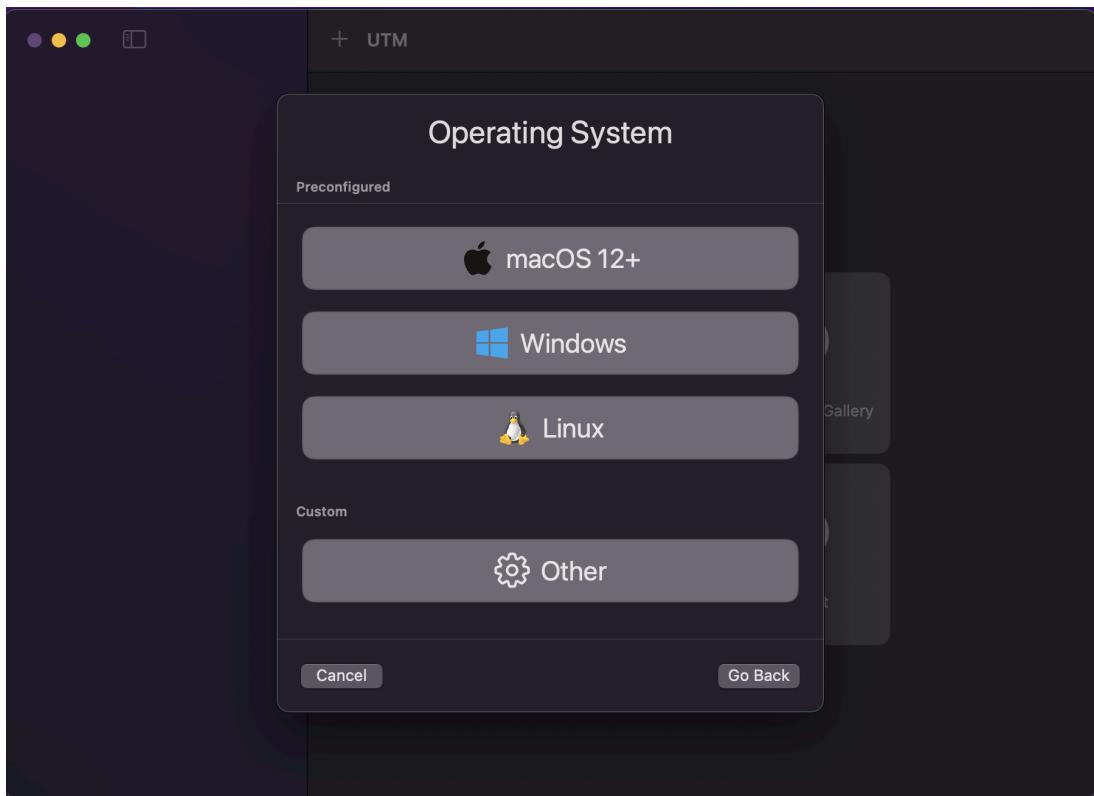
3. Download Ubuntu 22.04 Linux ISO Image

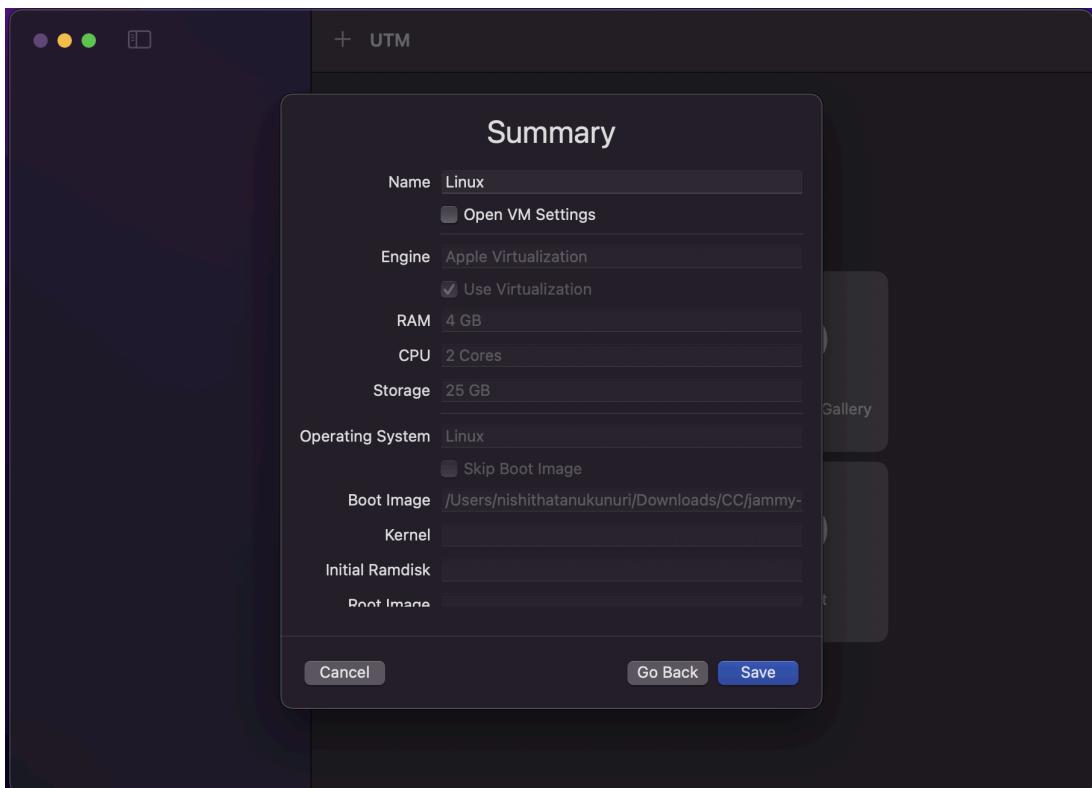
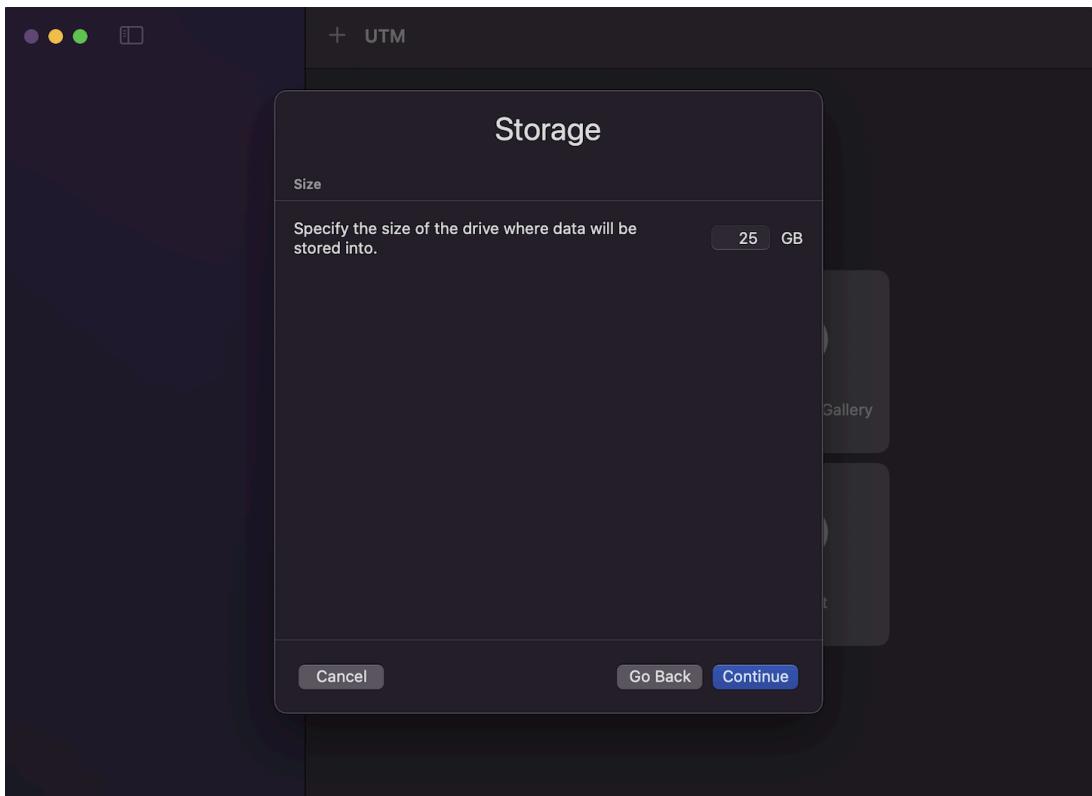


The screenshot shows the Ubuntu 22.04.3 LTS (Jammy Jellyfish) download page. At the top, there's a header with the Ubuntu logo and the text "ubuntu releases". Below the header, the title "Ubuntu 22.04.3 LTS (Jammy Jellyfish)" is displayed. A large orange banner with the text "Select an image" is visible. 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."

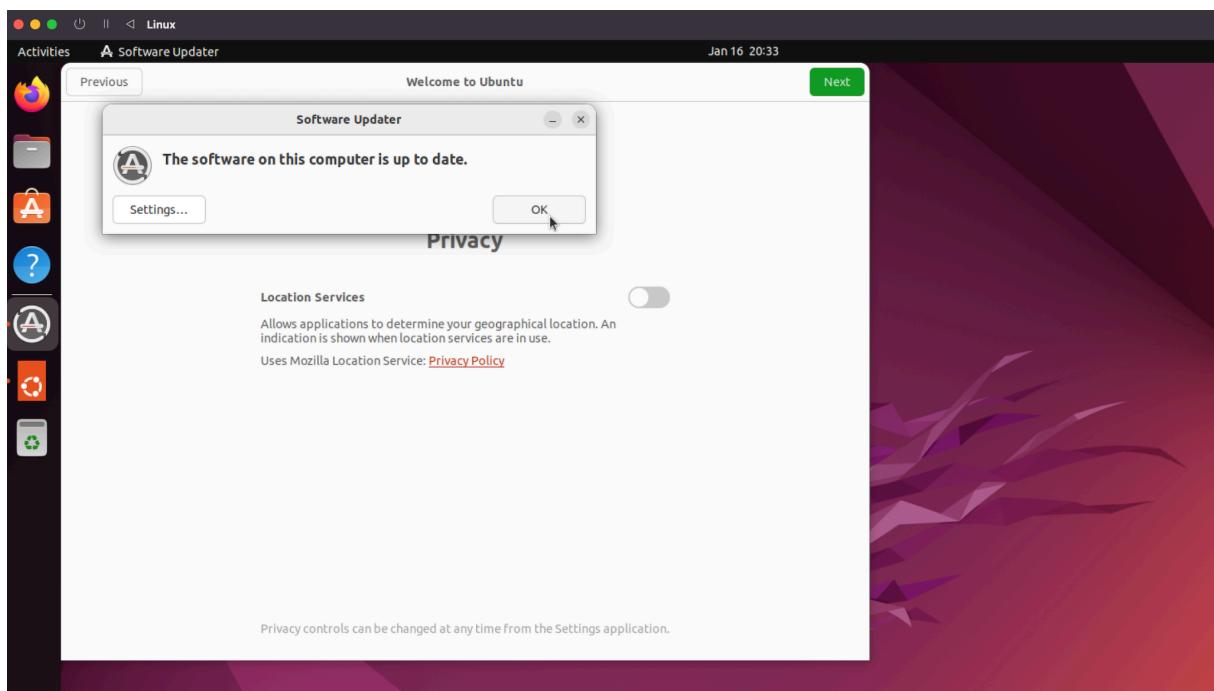
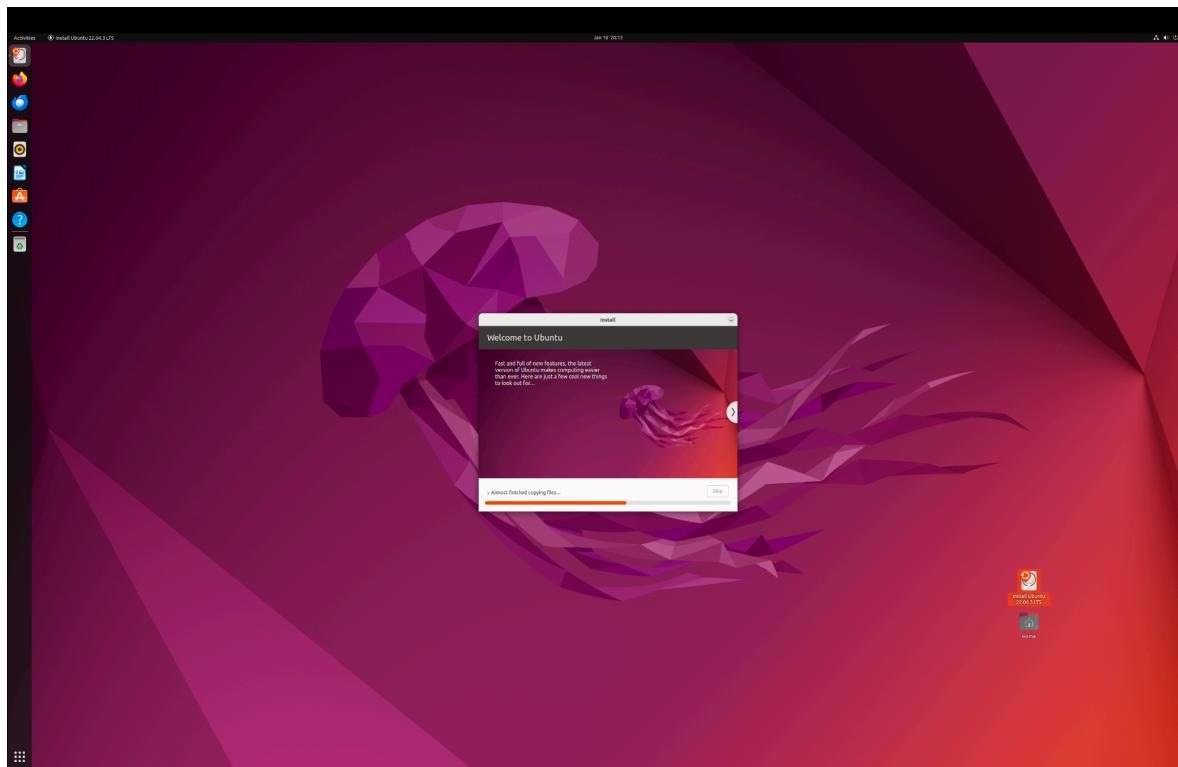
4. Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with Bridged Adapter





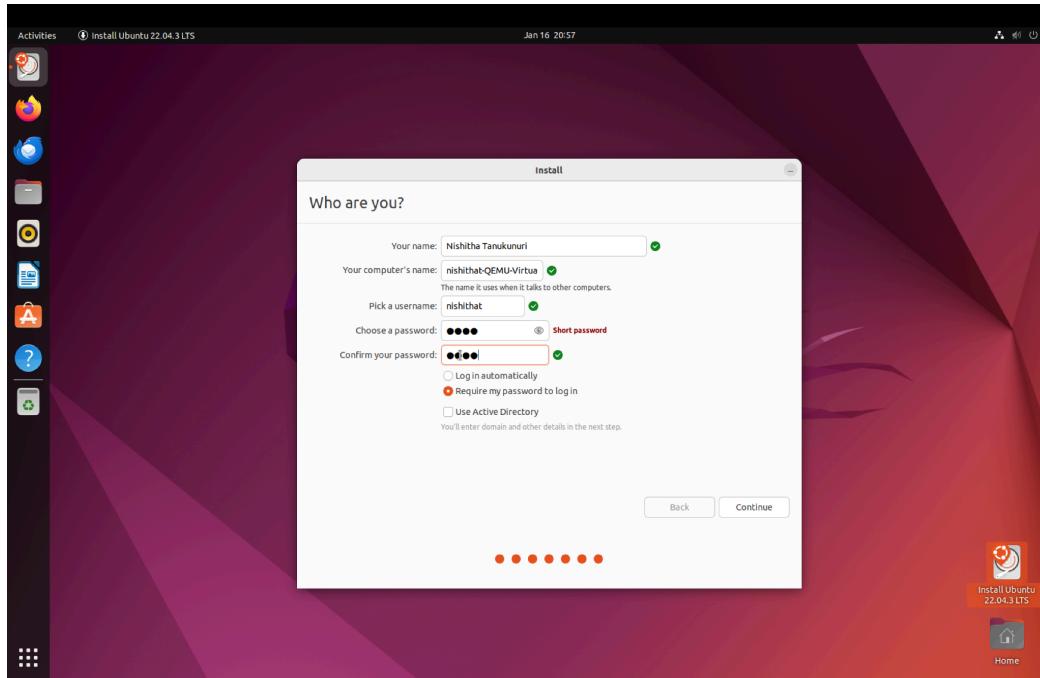


5. Install Linux from the ISO image

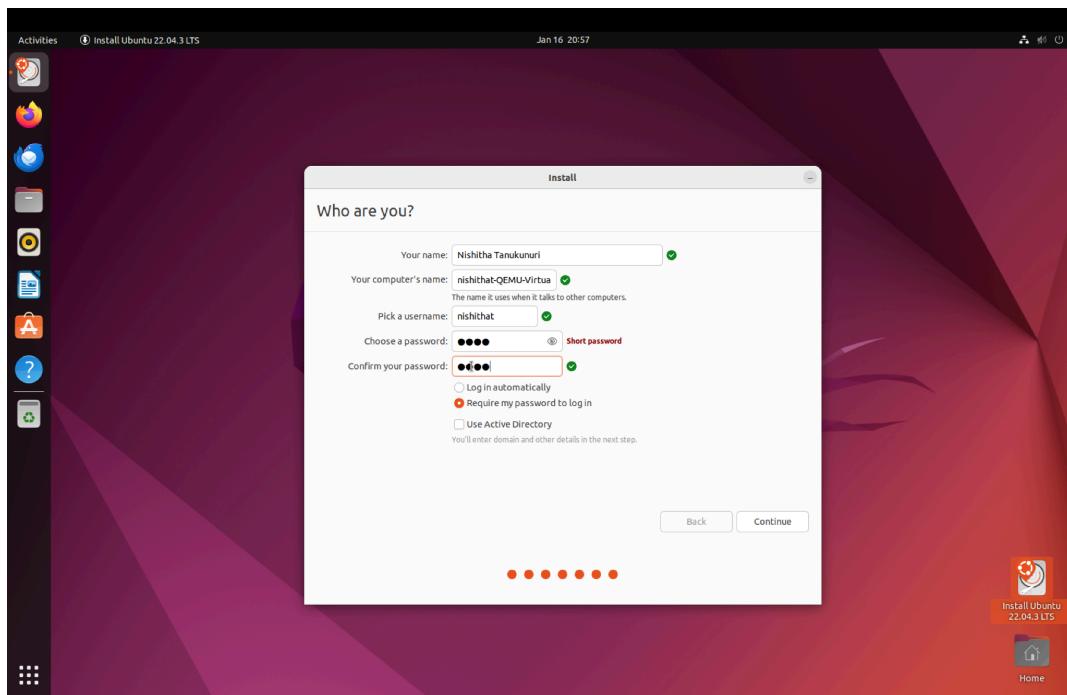


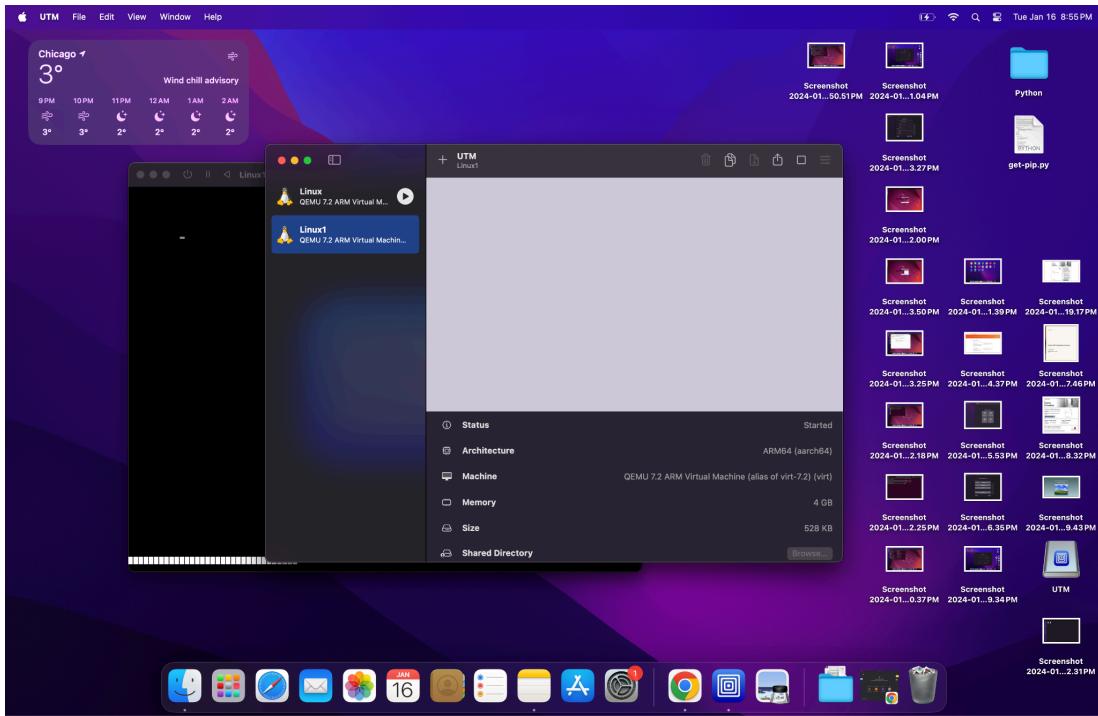
6. Create a user ID and password

For VM 1:



For VM 2:



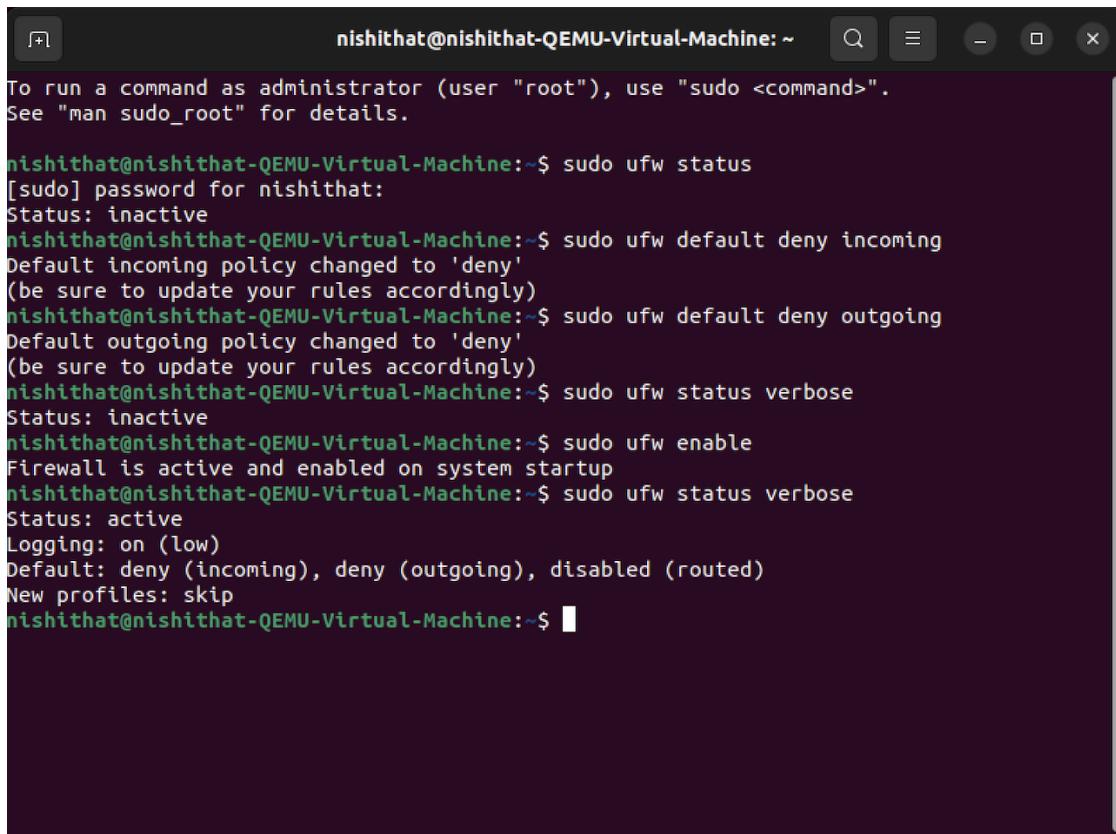


7. Turn on Firewall and block all ports

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 a dark background and light-colored text. The title bar reads "nishithat@nishithat-QEMU-Virtual-Machine: ~". The terminal output is as follows:

```
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

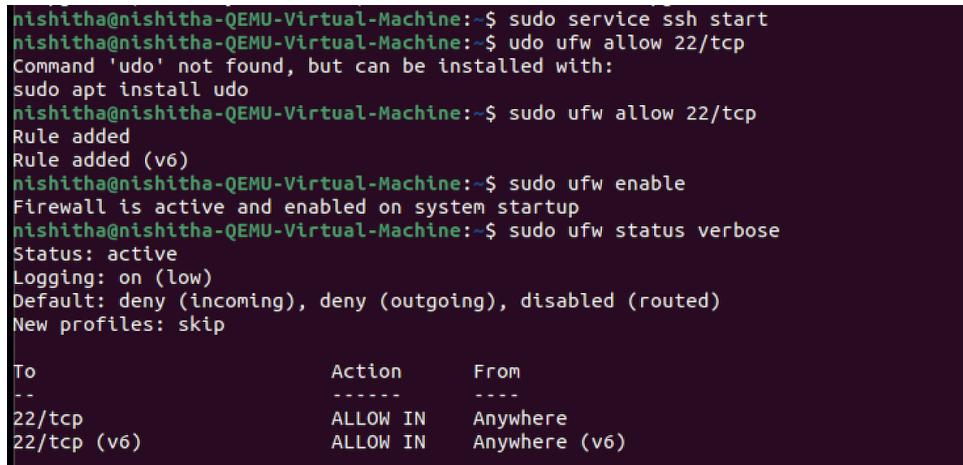
[nishithat@nishithat-QEMU-Virtual-Machine:~]$ sudo ufw enable
Firewall is active and enabled on system startup

[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:

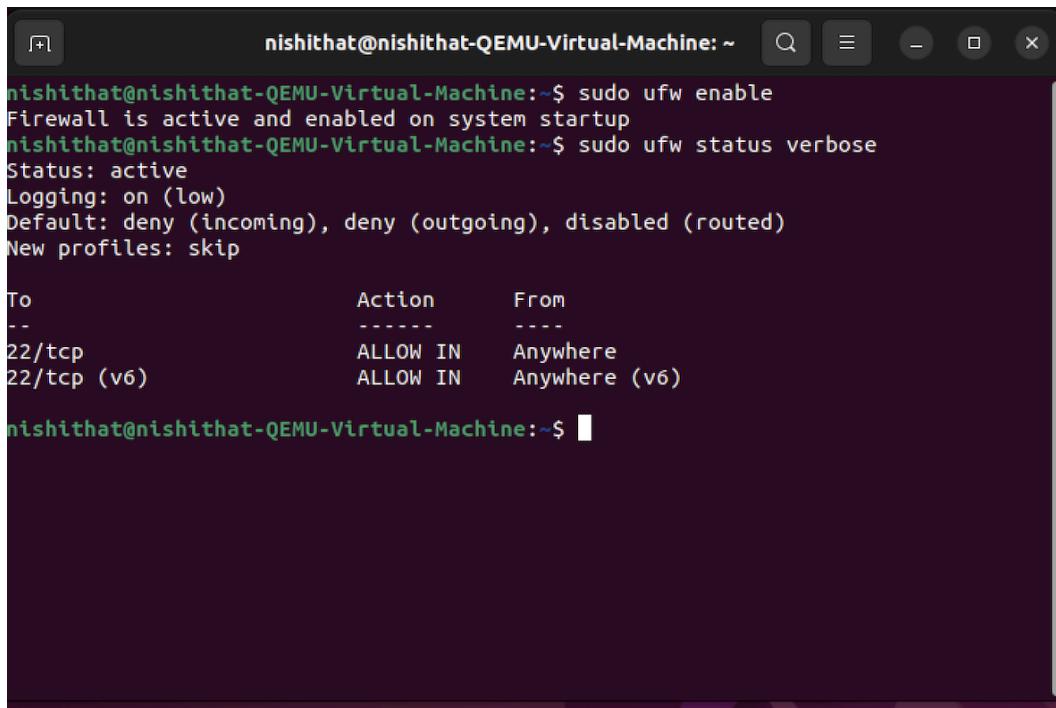


The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "nishitha@nishitha-QEMU-Virtual-Machine: ~". The terminal output is as follows:

```
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:



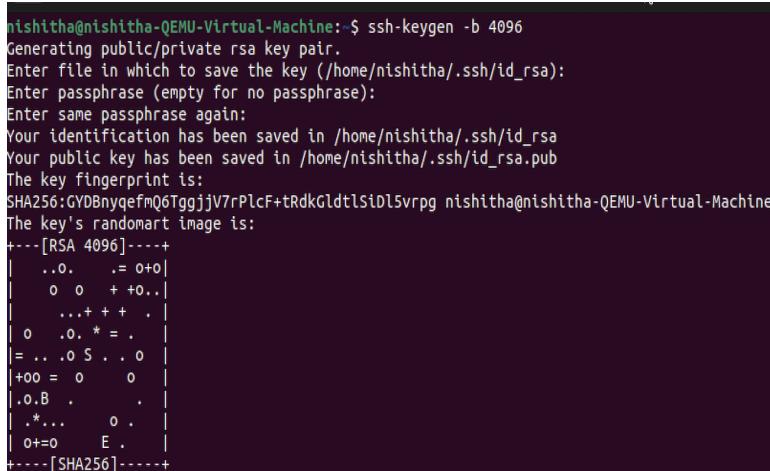
```
nishithat@nishithat-QEMU-Virtual-Machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
nishithat@nishithat-QEMU-Virtual-Machine:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

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:yuhXsWkC0KyS32Rnt7MikpvOOQQMJF5+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