

How To Make The Best Use Of Live Sessions

- Please login on time
- Please do a check on your network connection and audio before the class to have a smooth session
- All participants will be on mute, by default. You will be unmuted when requested or as needed
- Please use the “Questions” panel on your webinar tool to interact with the instructor at any point during the class
- Ask and answer questions to make your learning interactive
- Please have the support phone number (US : 1855 818 0063 (toll free), India : +91 90191 17772) and raise tickets from LMS in case of any issues with the tool
- Most often logging off or rejoining will help solve the tool related issues

COURSE OUTLINE



MODULE 1

INTRODUCTION TO LINUX

INSTALLATION AND INITIALISATION

USER ADMINISTRATION

BOOT AND PACKAGE MANAGEMENT

NETWORKING

LINUX OVERVIEW AND SCRIPTING

LINUX FOR SOFTWARE DEVELOPMENT

SECURITY ADMINISTRATION

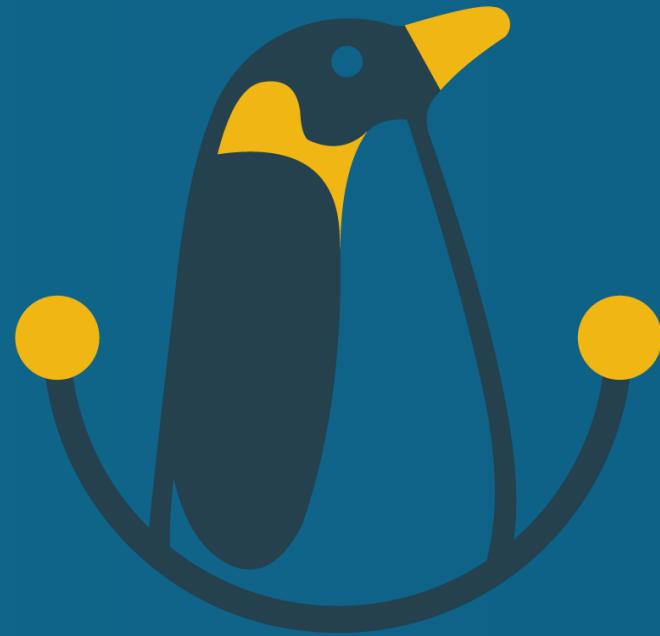
Objectives

After completing this module, you should be able to:

- Understand the History of Linux
- Differentiate between Linux and Unix
- Learn the Architecture of Linux
- Choose the appropriate Linux Distribution
- Understand Legal Aspects associated with a Software
- Install Ubuntu on VirtualBox



edureka!



Introduction To Linux



Why Do We Need Linux OS?



Facebook For Social Stability And Support



- Facebook's platform is built on a modified version of CentOS, with certified support from Red Hat
- CentOS, Distribution of Linux, provides the stability to the company which deliver a consistent user experience, while Red Hat comes with proprietary software packages and enterprise support

DreamWorks For Rendering Movie Magic



DREAMWORKS

- The production powerhouse behind hit films such as Madagascar, Kung Fu Panda and Shrek, the animation sensation that made the company a monster
- According to Red Hat, nearly every artist at DreamWorks use Linux on their desktop and server to help streamline production processes, increase efficiency and reduce cost

Google For Doing Google Stuff



- Google has its own customized Linux distribution called “**Goobunta**”, which is modified version of Ubuntu
- **Goobunta** forms the core of the company’s search, advertising, and cloud computing platforms
- Google prefers Linux for its superior out-of-the-box security, remarkable performance, and unmatched flexibility

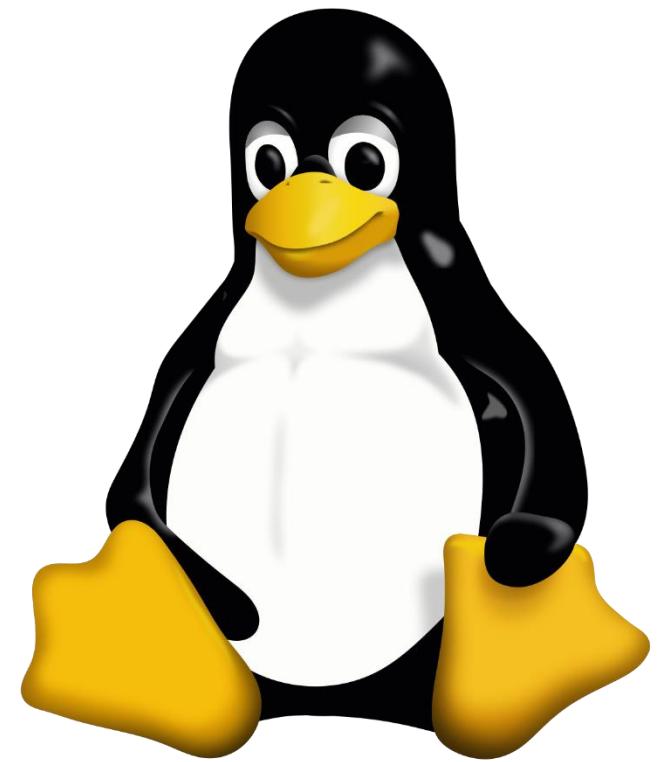
Munich For System Liberation



- The city of Munich(Germany) recently completed a 10-year project by migrating nearly 15,000 desktops on its network through “LiMux”, again a customized distro of Linux
- Linux makes an ideal solution with a secure architecture, cost effectiveness and flexibility that allows different offices to seamlessly share information

What Is Linux ?

Linux is an **Open Source Operating System** modelled on **Unix**, and developed in C language





Open Source

01

A software becomes open source if its source code is freely available

02

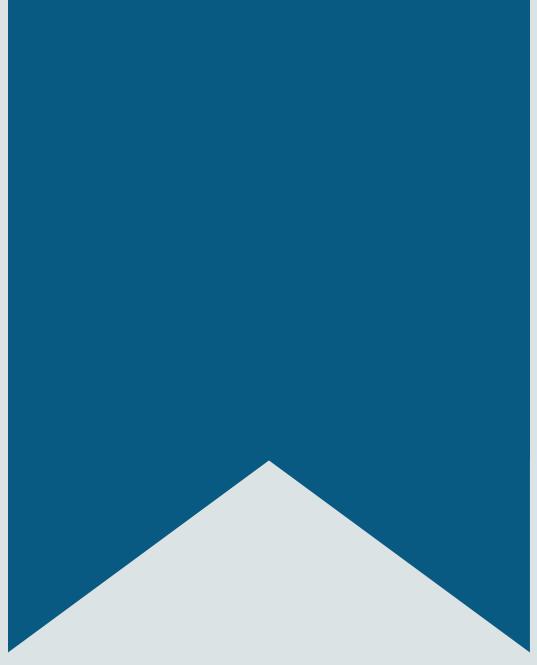
The free software movement was started in 1983 and 1998. Some developers coined the term “Open-Source” to make it less ambiguous and everyone adapted to it

03

Open source projects are generally a collaborative effort by multiple sets of developers to enhance the product and allow others to get benefit of it

04

The owner may put restrictions on usage, modification and distribution by various licensing, but it should be available to study for everyone



History Of Linux

History Of Linux

1969 and 1970

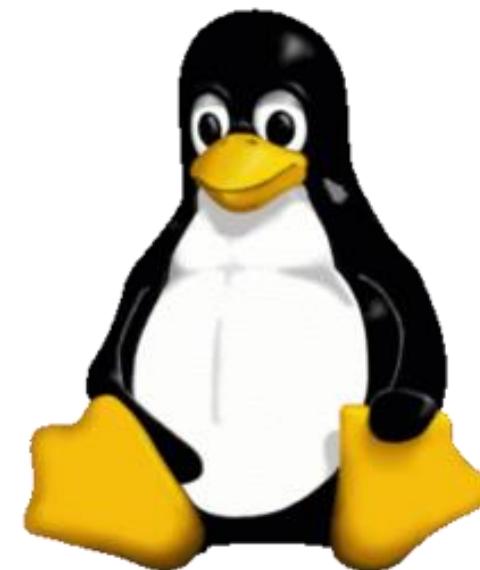
Unix OS was conceived, implemented and released by Ken Thompson and Dennis Ritchie

1977

BSD (Berkeley software Distribution) contained Unix code that was developed by UC Berkeley

1983 and 1986

Stallman started a GNU project for Unix like OS, which was free for copying and modification. In 1986, Maurice J published the design of Unix OS



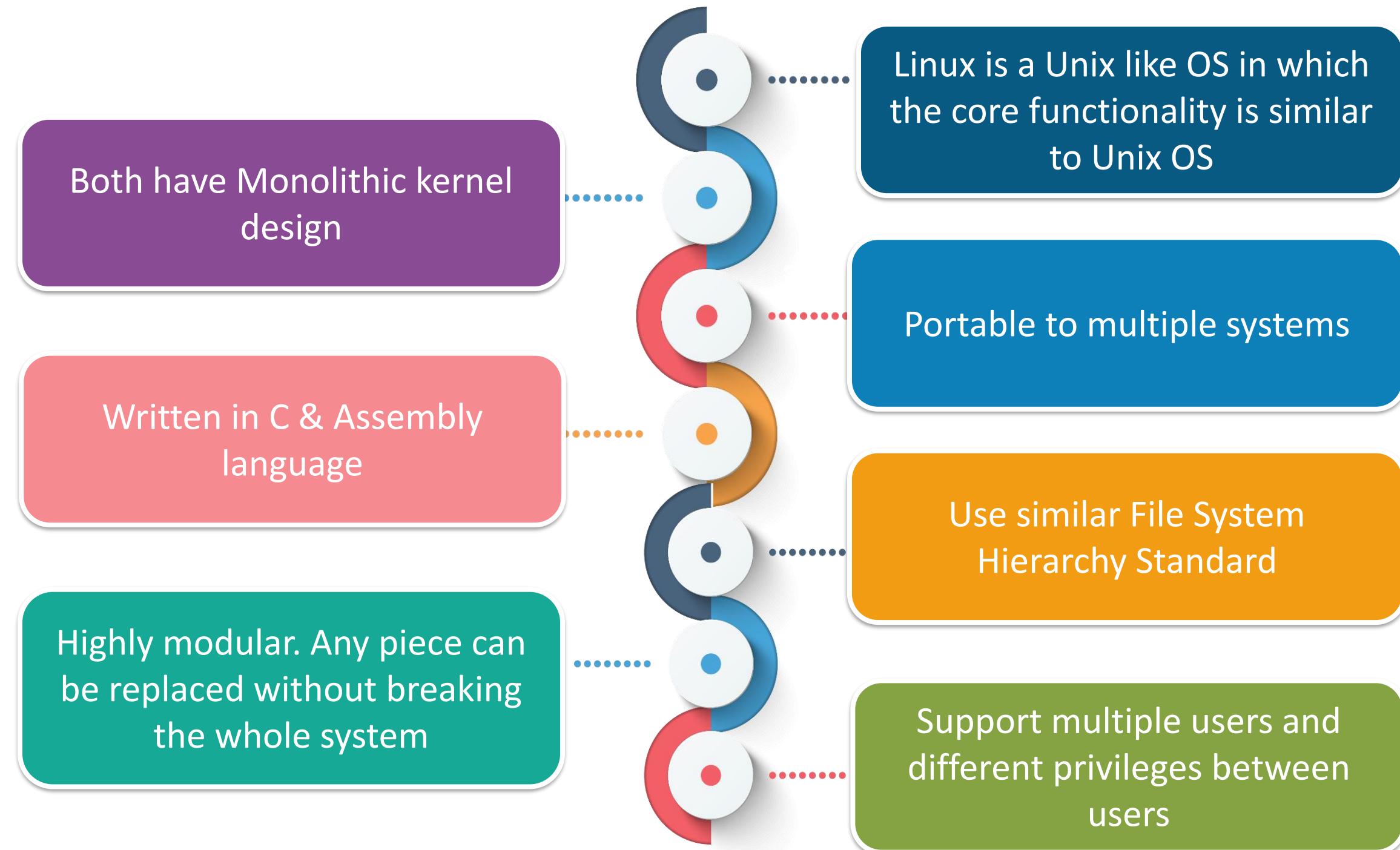
1991

Torvalds created a Linux Kernel in 1991. Linux Kernel along with GNU Tools became Linux Operating System

1987

MINIX, which similar to Unix was developed by Andrew Tanenbaum, where copying of code was allowed

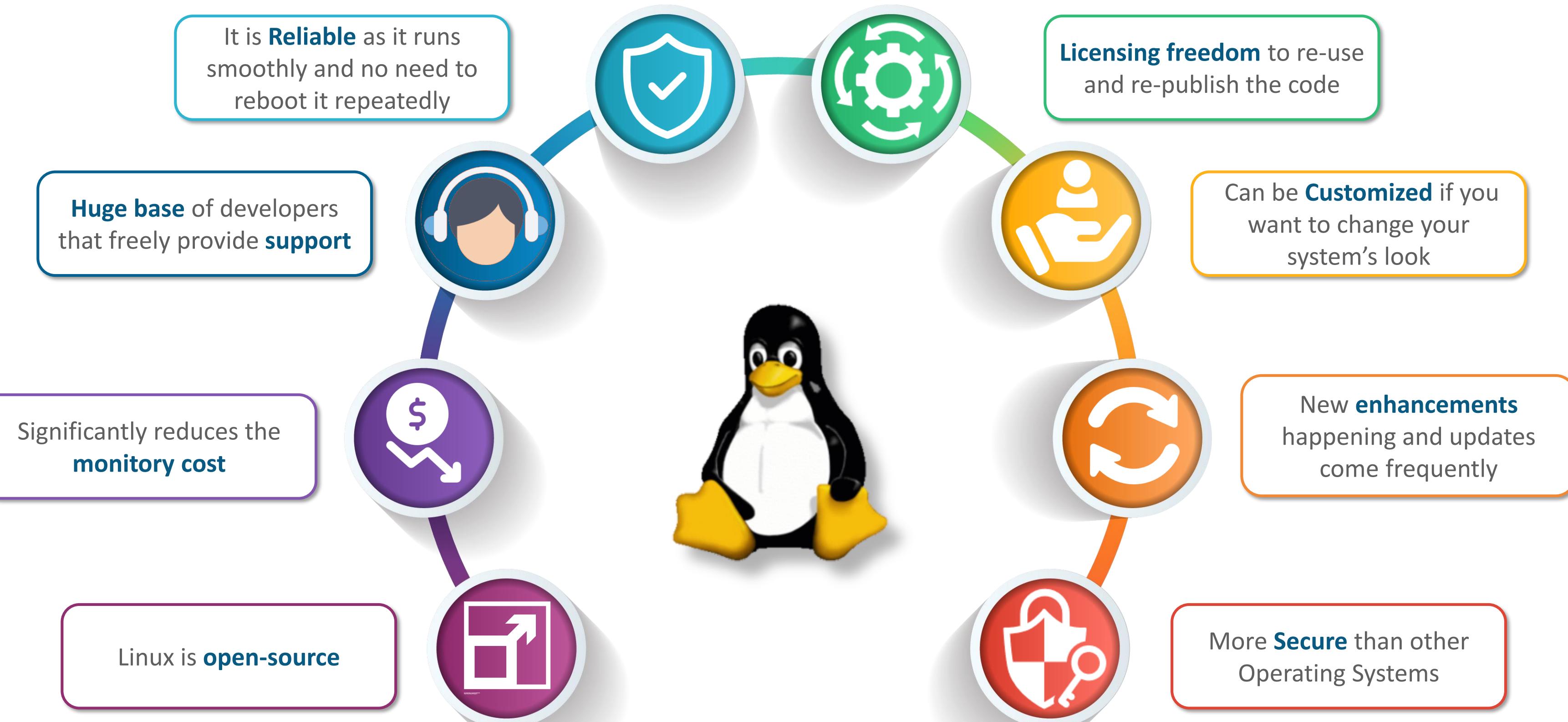
Relationship Between Unix And Linux



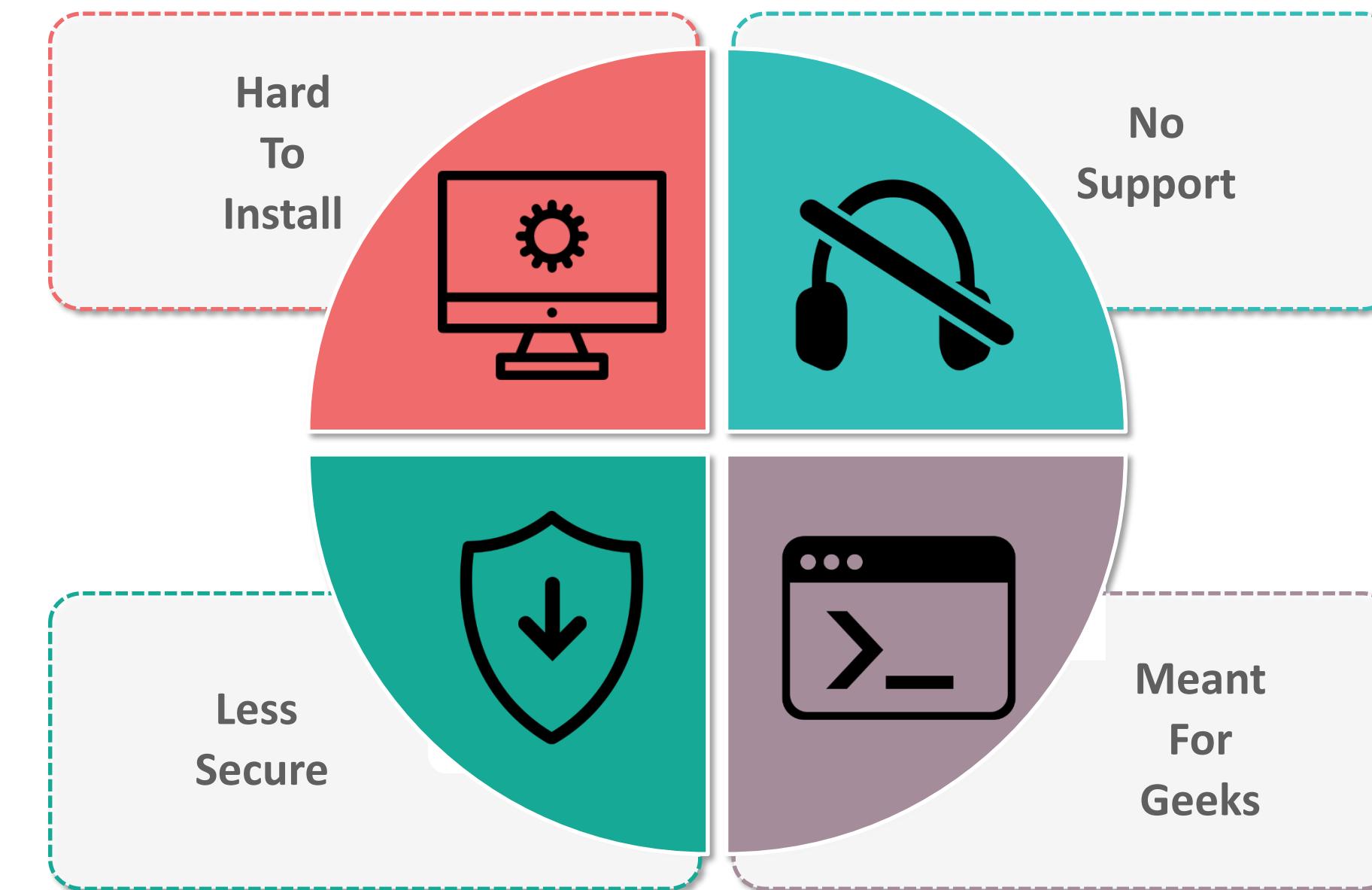
Linux Vs Unix

Parameters	Linux	Unix
Cost	Free	Not Free Different vendors have different prices
Flexibility	Yes	Not Flexible Compatible with lesser hardware
Source Code	Available	Not available As it is not a freeware
Installation	Economical	Uneconomical Unix requires specific hardware
Community	Wide network of developers	Limited commercial developers
Support	Depends on forums and community for support	Vendors usually provide technical support for commercial Unix
Bug Fixes	Bug fixing is much faster in Linux than in Unix	

Features Of Linux

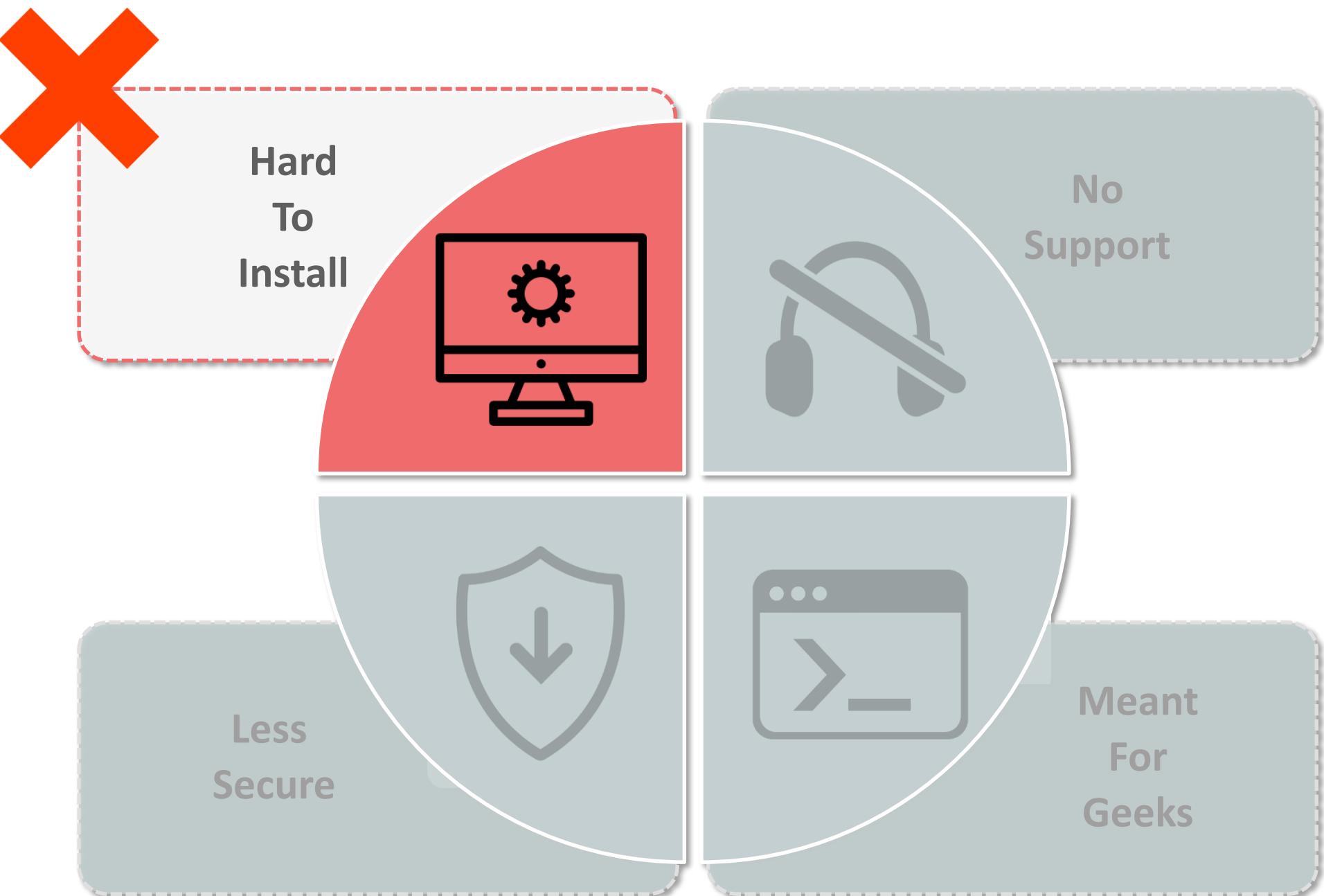


False Myths Around Linux



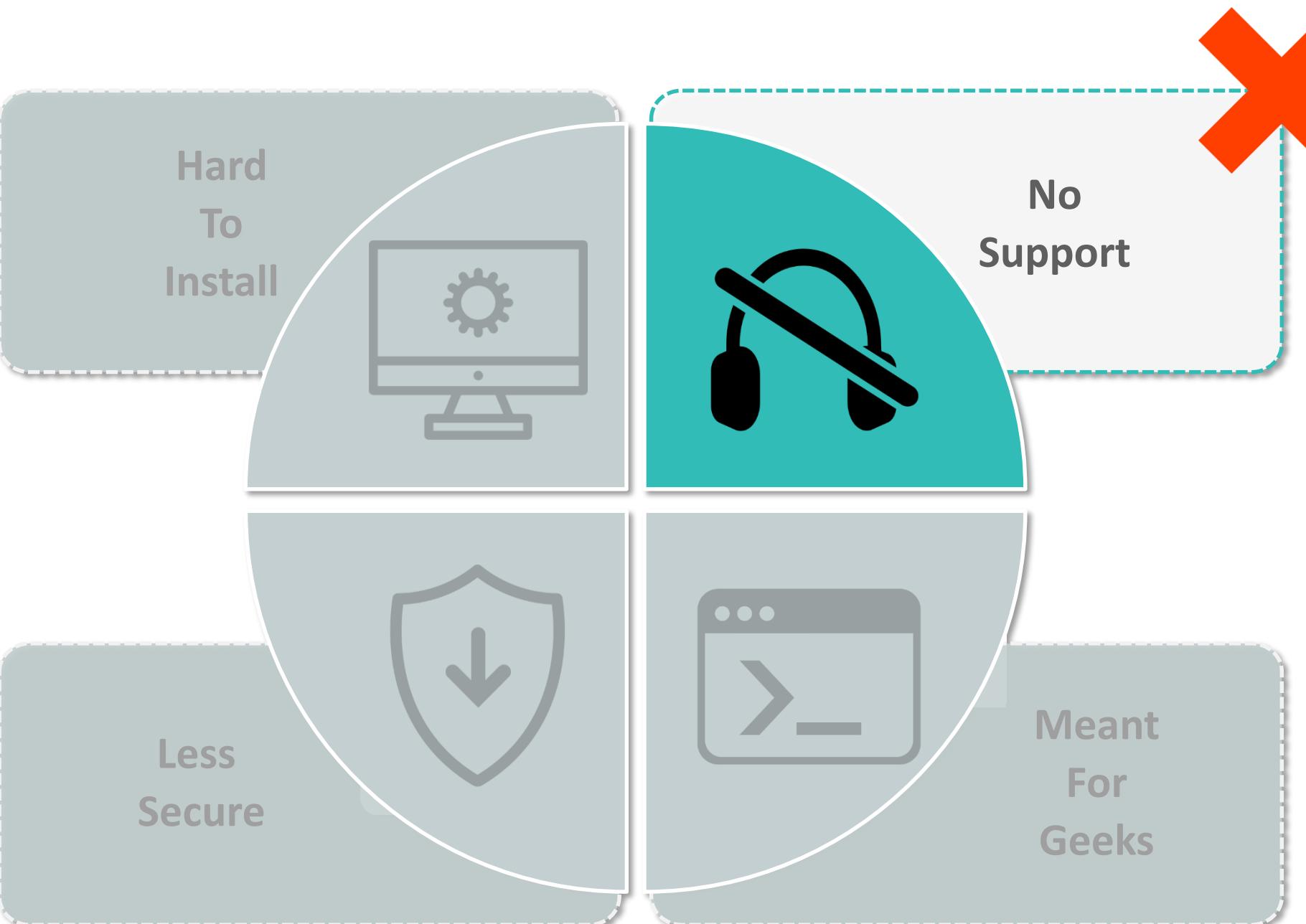
False Myths Around Linux

Nowadays, Linux is readily available and requires fewer steps to install than Windows. It also has a user friendly description for each step



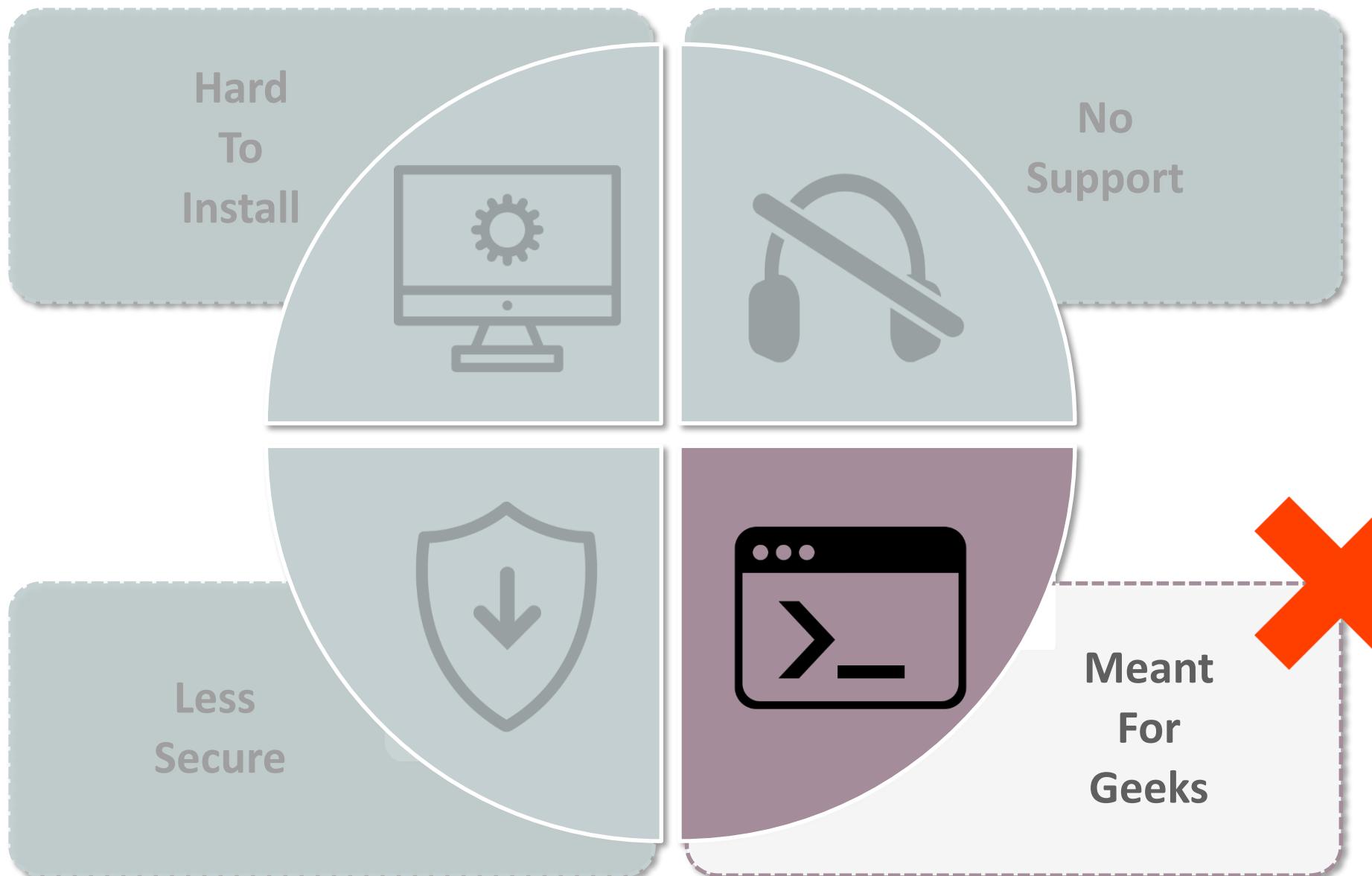
False Myths Around Linux

It has the largest community of developers, who not only provide solution for new issues coming in but also help with everything



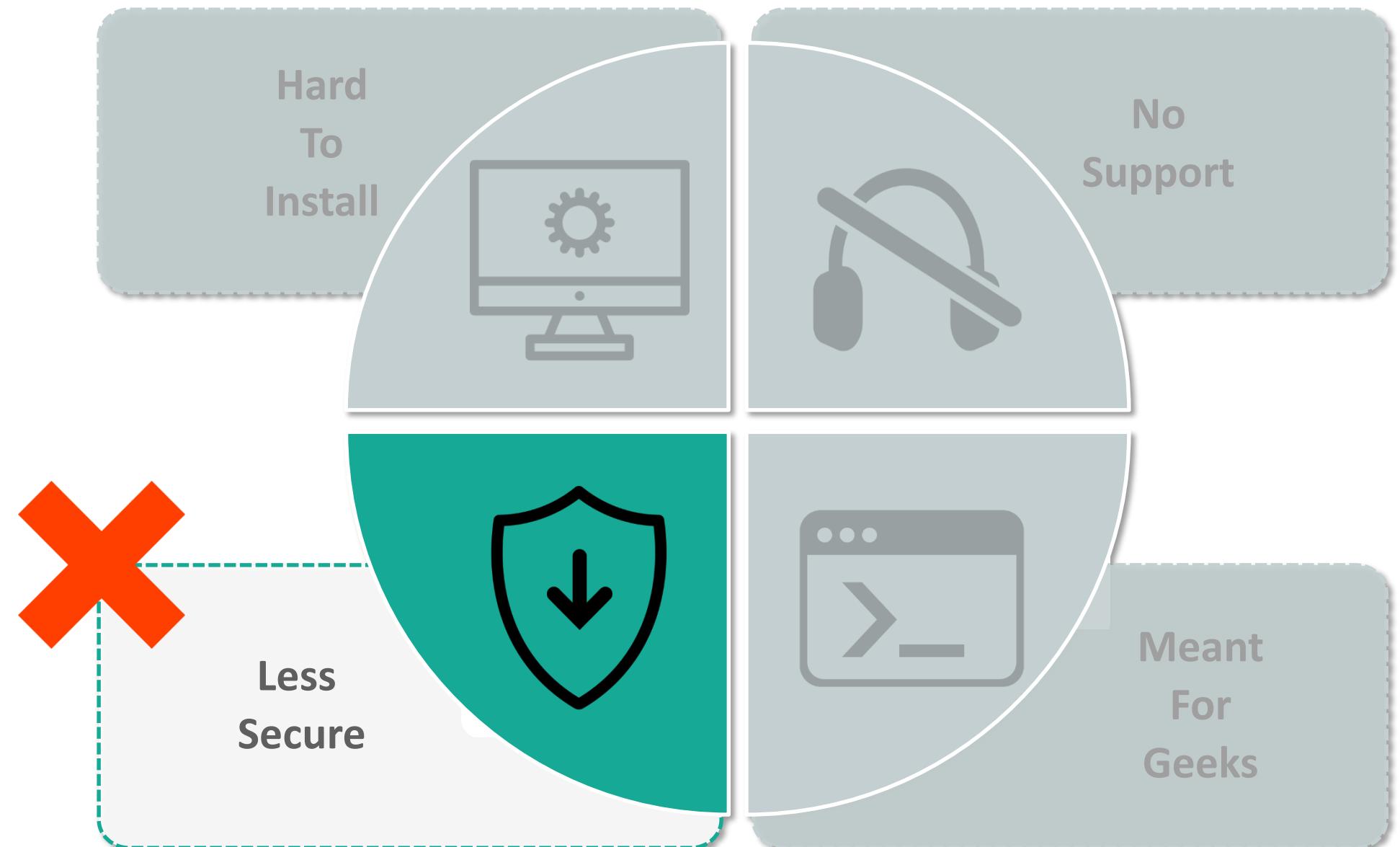
False Myths Around Linux

You would not require coding skills to install or use Linux because it has Command Line Interface which is a shortcut to run things in Linux

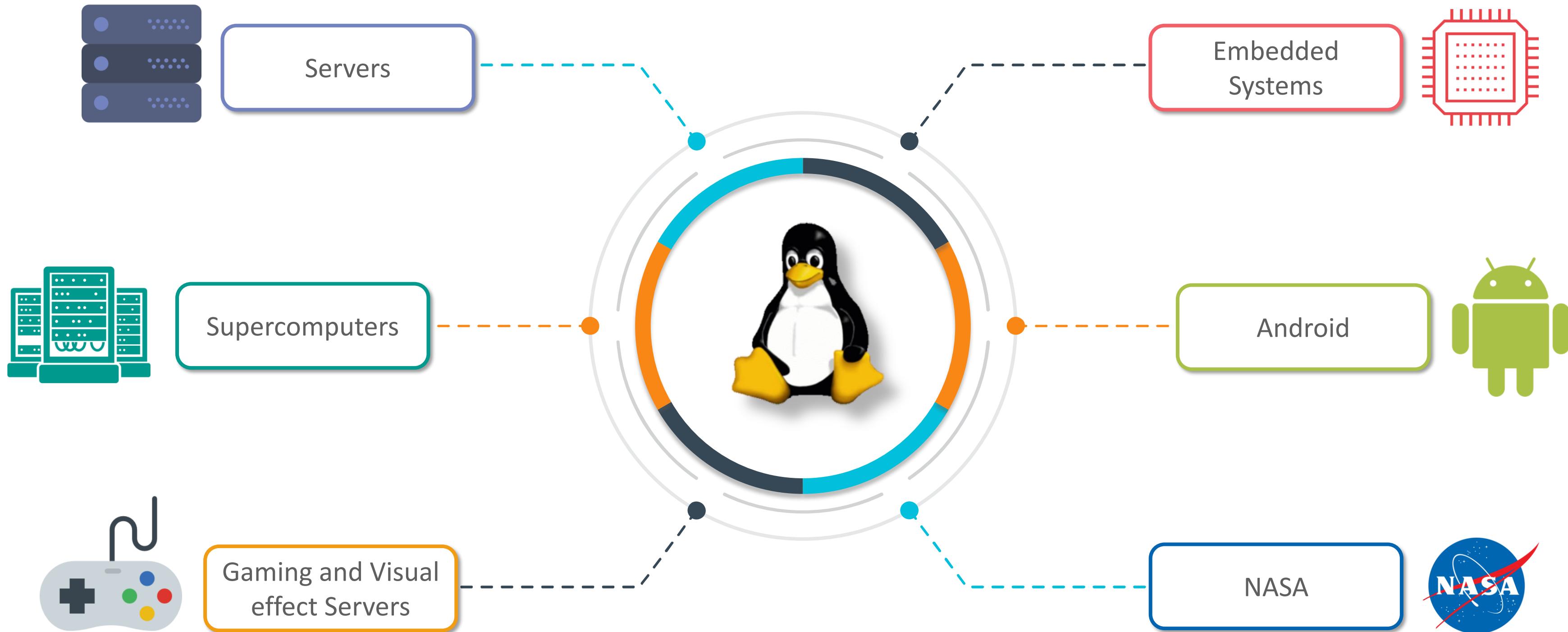


False Myths Around Linux

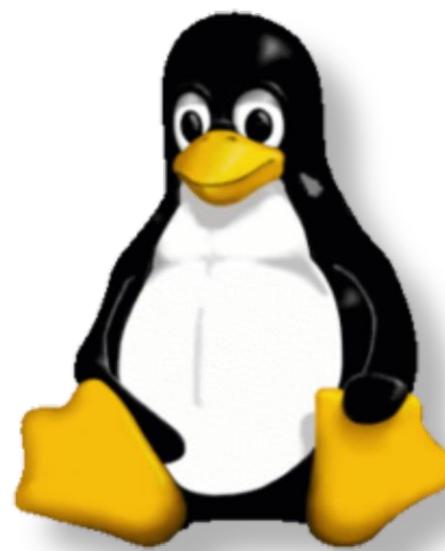
Linux is far more secure than other OS, but one needs to know basic things before downloading new software to not tamper with its security



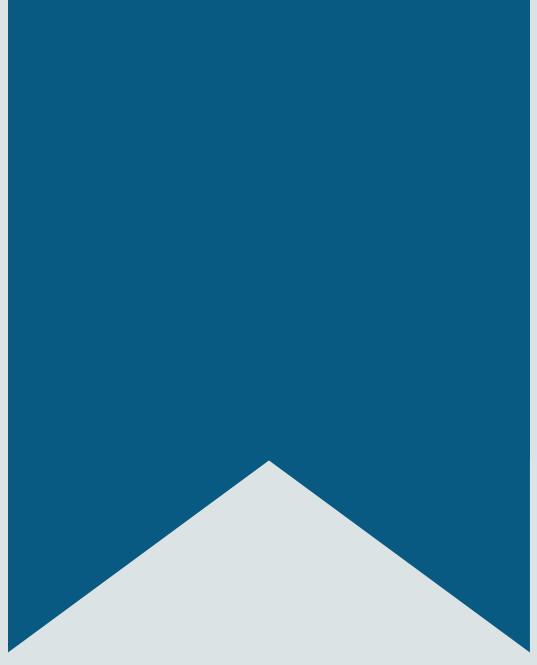
Where Linux Is Used?



Components Of A Linux OS

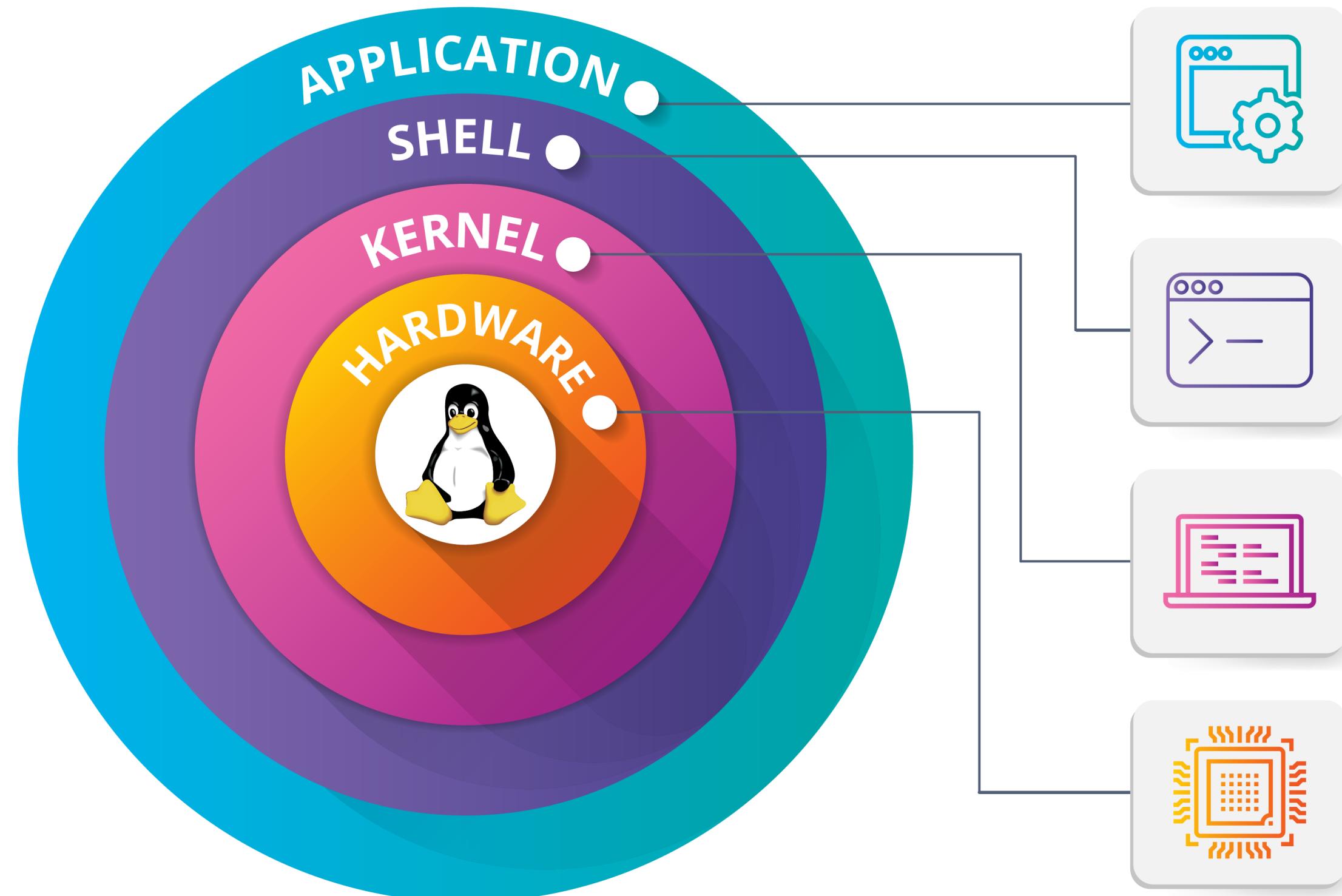


- 1 **Init Program:** The first program, which is responsible for the initialization of the system. Ex: sysvinit, systemd
- 2 **Bootloader:** A program which loads the operating system when the system is turned on. Ex: GNU Grub, Syslinux
- 3 **Software Libraries:** Set of programming code used to develop and design software programs and applications.
- 4 **Package Management System:** A collection of tools for Installation, Deletion, Configuration, and Upgradation of software. Ex: dpkg, RPM
- 5 **Other Interface and application:** Various applications and interfaces are used for various task performed by user.



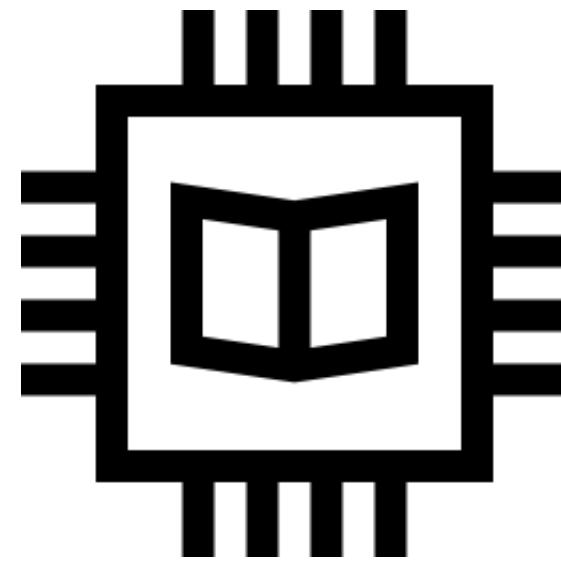
Architecture Of Linux OS

Architecture Of Linux OS



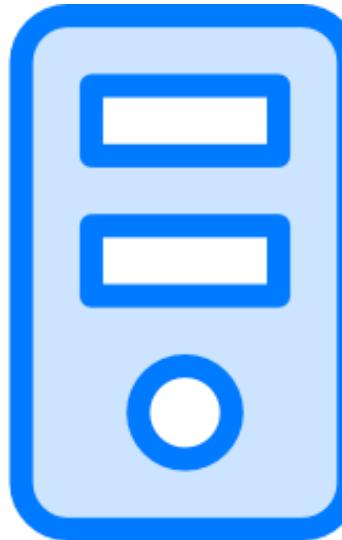
Hardware

Hardware part of architecture includes all the peripheral devices. For Example: CPU, RAM, Hard Disk Drive,etc



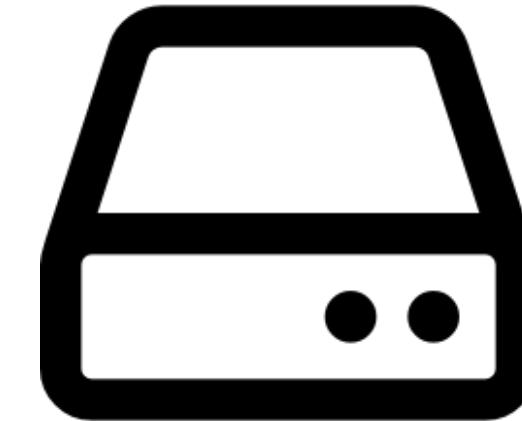
RAM

It is volatile memory space that stores the data which is directly accessed by CPU



CPU

It is a Electronic Component that carries out the instructions of computer program



Hard Disk Drive

It is non-volatile memory, where the Operating System is stored

Kernel

The kernel is the Interface between the Applications and the actual process done at the Hardware level

Tasks performed by the Kernel

Resource Allocation

Manage the computer's resources and allow other programs to run and use these resources

Security Management

Provides security and protection from faults and malicious behaviours

Process management

Allows the execution of applications and support them with features such as hardware abstraction

Device Management

Maintains a list of available devices and allow drivers to physically access their devices

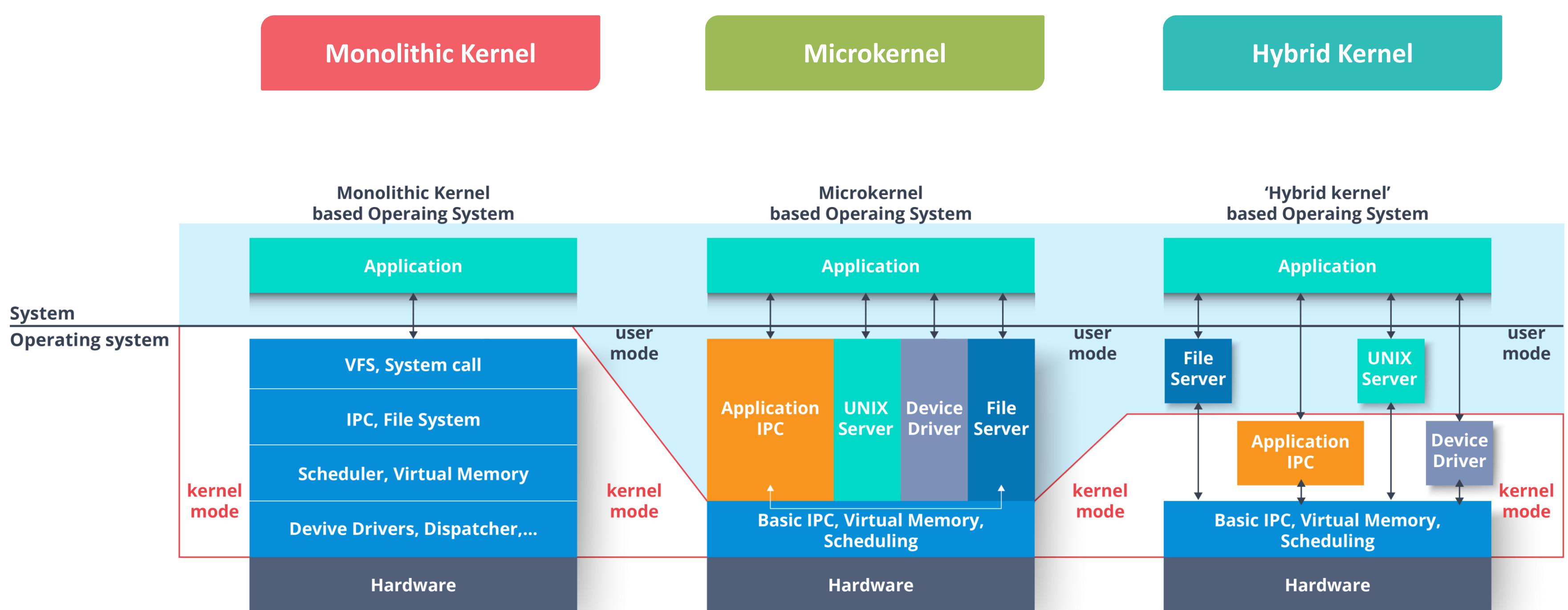
Scheduling

Gives every program specific amount of time and can switch from one process to another

Memory Management

Allows processes to safely access the memory according to their requirements

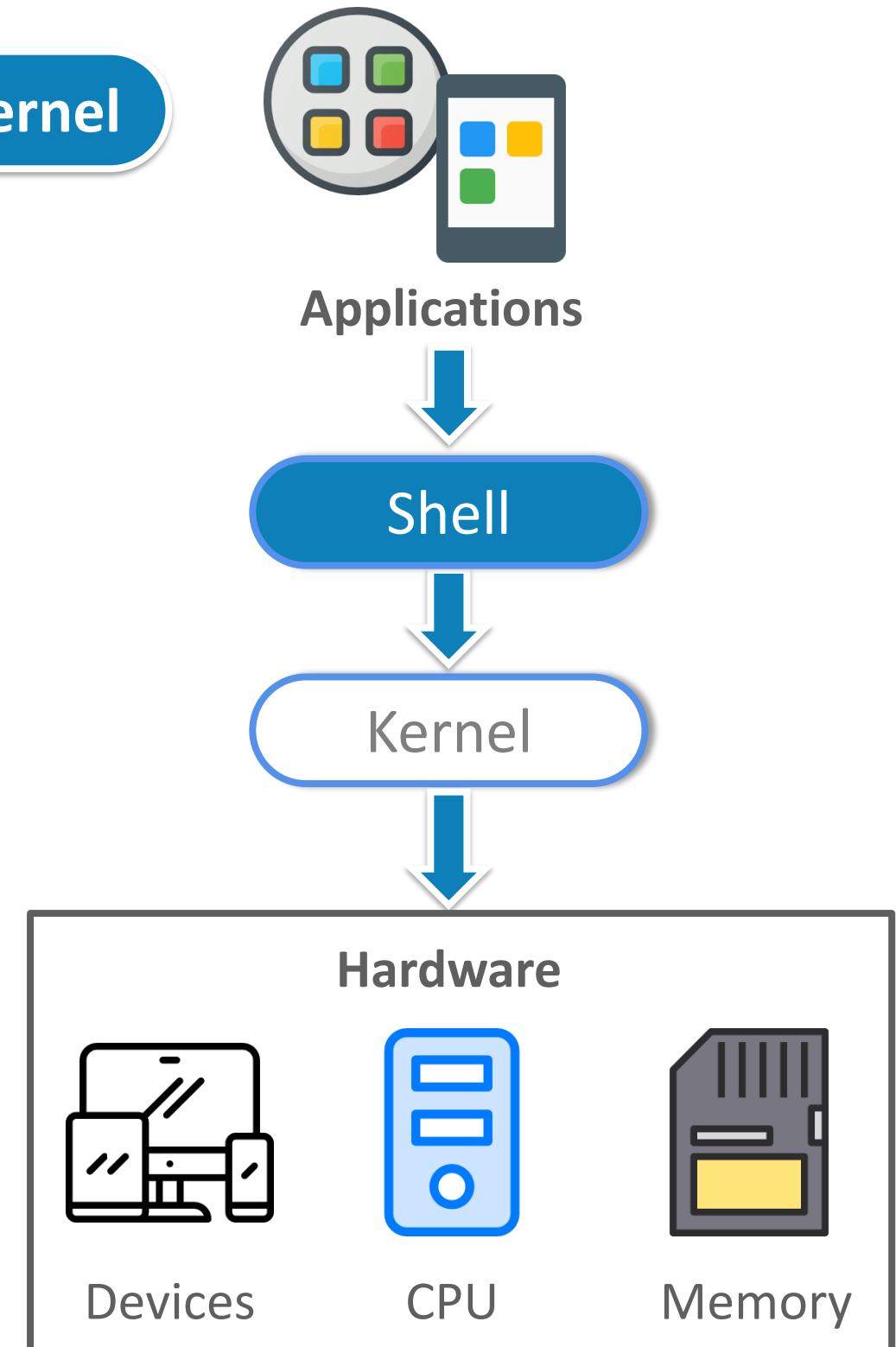
Types Of Kernel



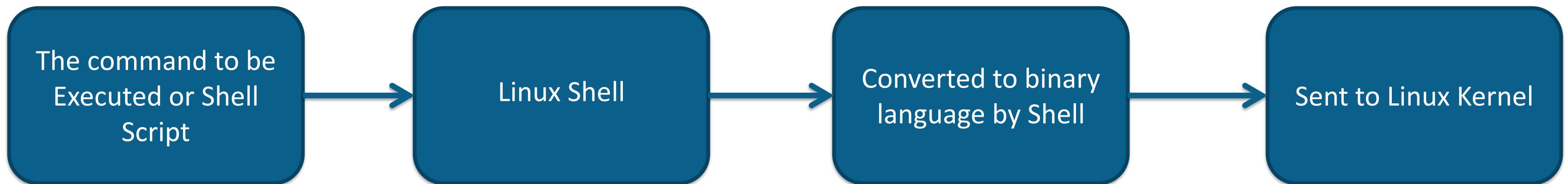
Shell

The shell is the **interface** which takes **user-command** and **sends it to the kernel**

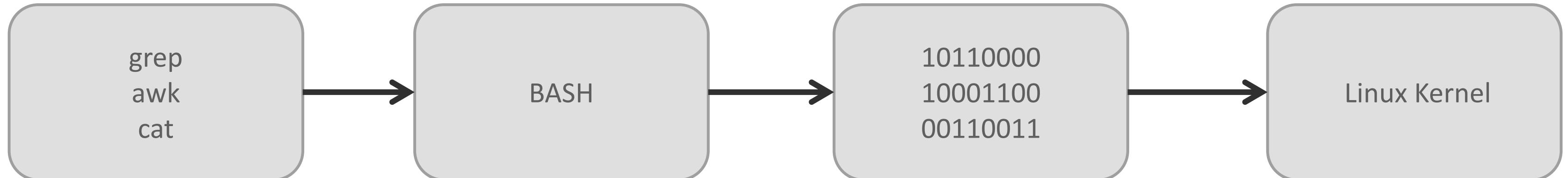
- Shell has got its own programming language and you can combine multiple commands in a single script
- The shell takes the command in human readable format and provides it to kernel in binary language
- The user generally interacts via shell, but direct interaction with hardware is also possible
- The first shell created was **sh** for Unix systems. Linux still provide support for **sh** shell, but **bash** is more popular within the Linux users



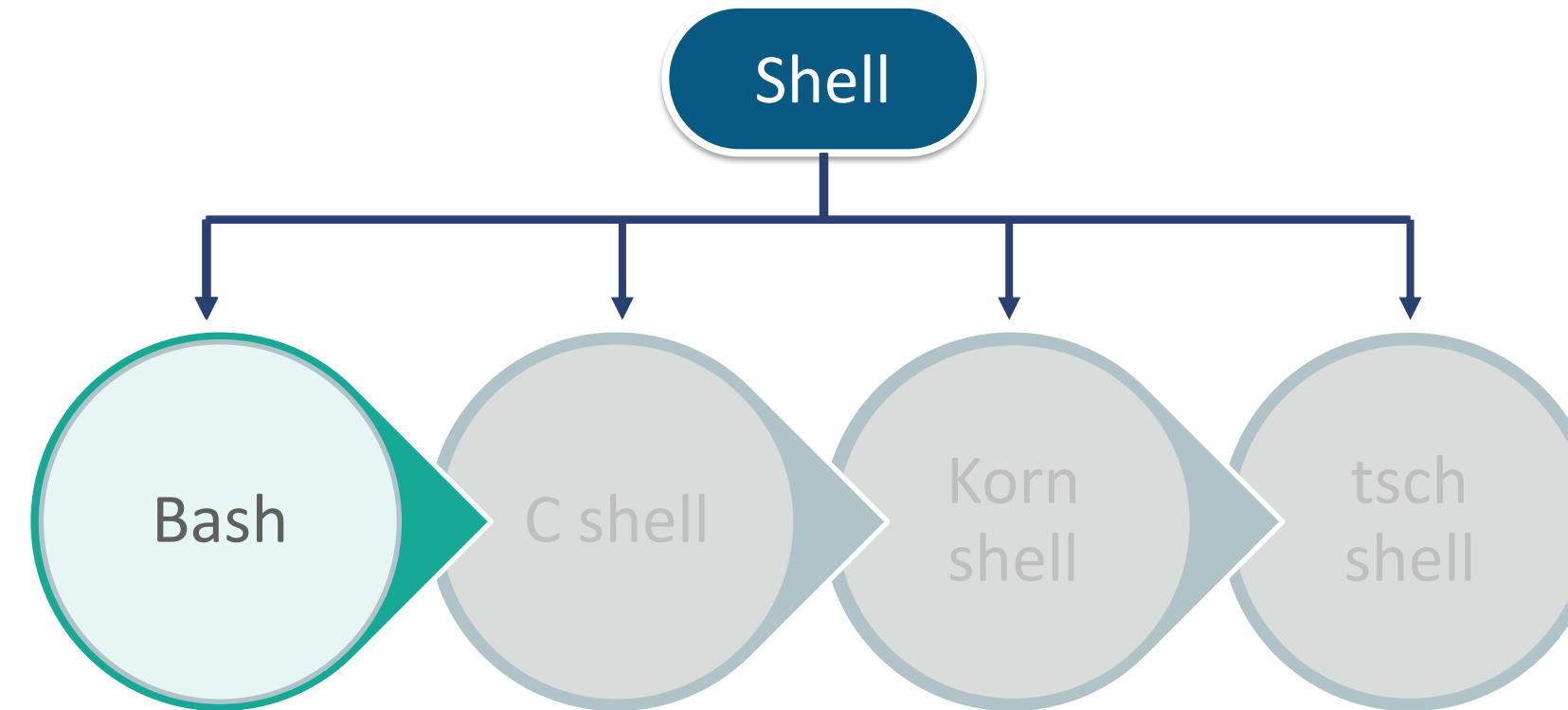
How Shell Works?



Example

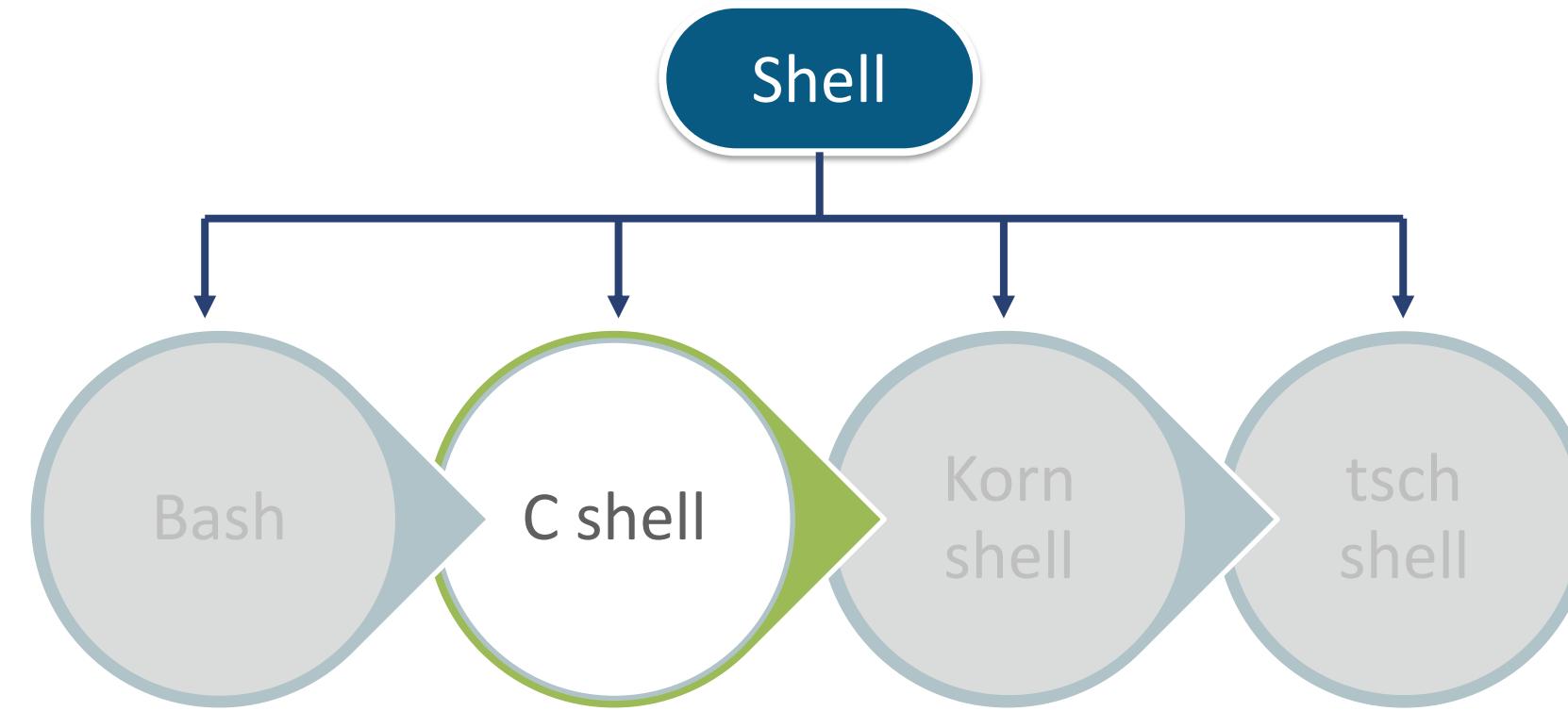


Types Of Shell



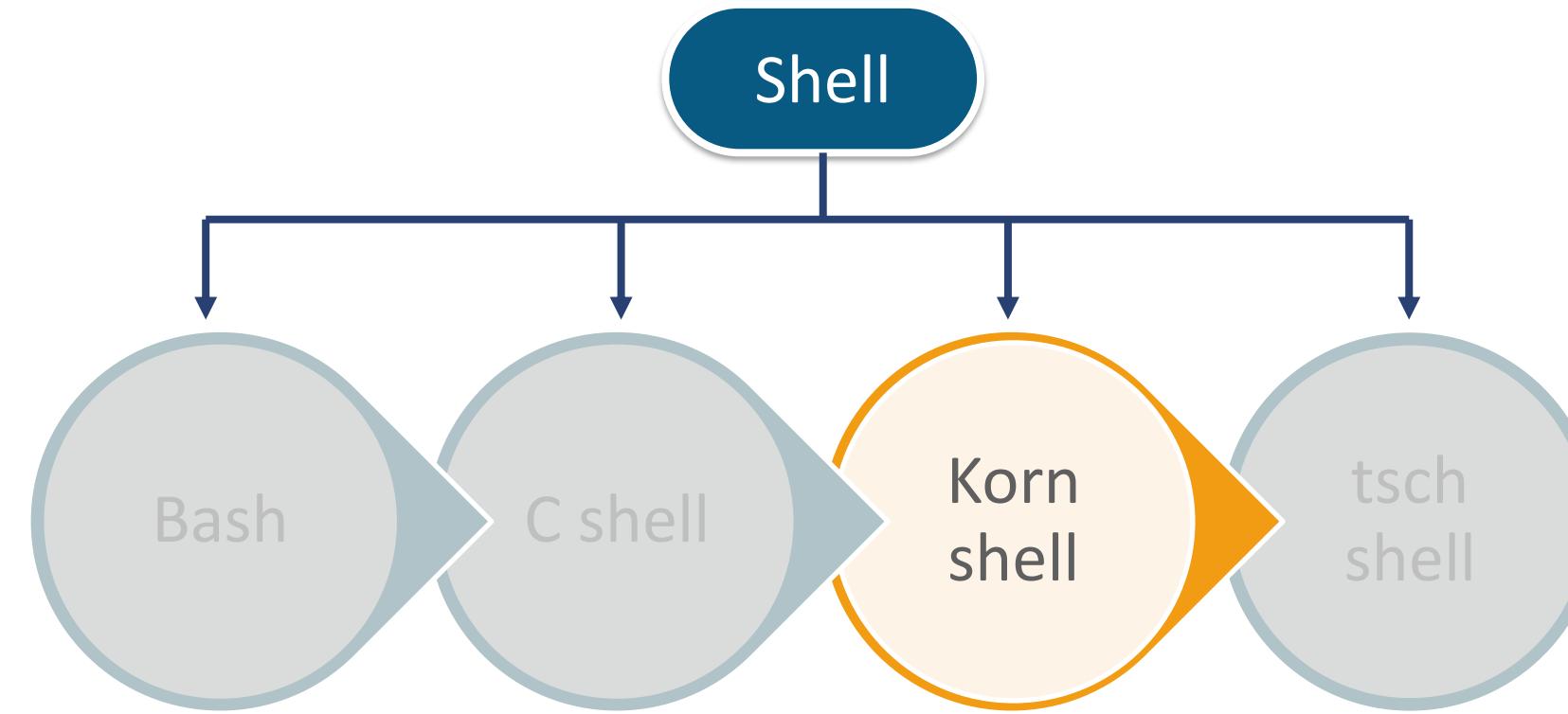
- Bourne-Again-shell was written as part of the GNU project to enhance the functionality of **sh** (Bourne-shell)
- Some enhancements like command completion and complete command history are done
- It supports all **sh** functionalities and has its own script for startup
- It can perform integral calculations without invoking any external process
- One can enter “#! /bin/bash” at the top of the file to tell Linux to run with bash interpreter

Types Of Shell



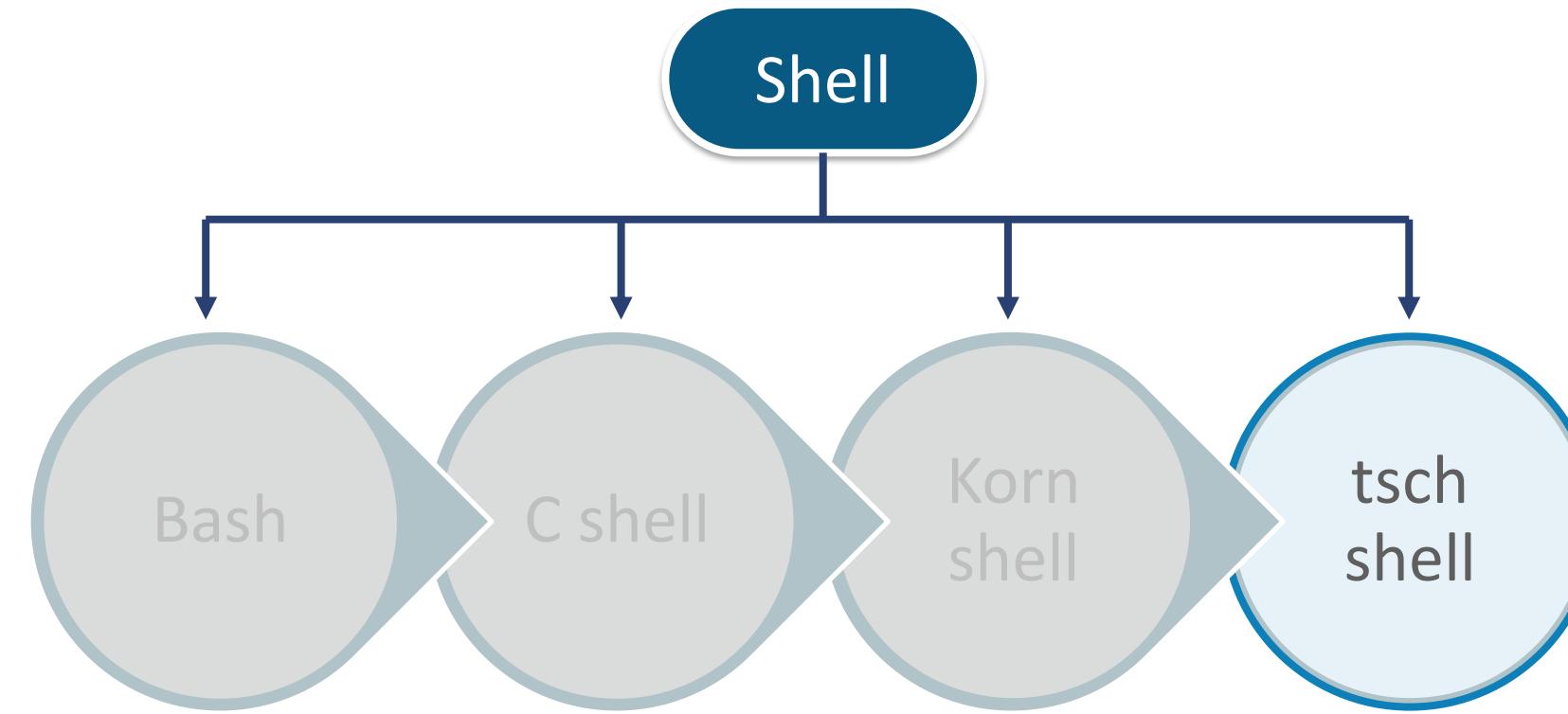
- It was written by Bill Joy and is widely distributed with BSD Unix
- Syntax and control statement are modeled and designed like C
- It is executed in a text window, which allow users to provide commands
- It can also read commands from the script
- It is linked to tcsh shell which is an improved version

Types Of Shell



- Korn shell includes C shell's major advantages and its own features
- It has command line editing which allows you to use vi or Emac style editing commands
- It enables common programming tasks to be done cleanly without creating extra processes
- A feature like debugging primitive make it possible to write tools that help the programmer to debug their shell code

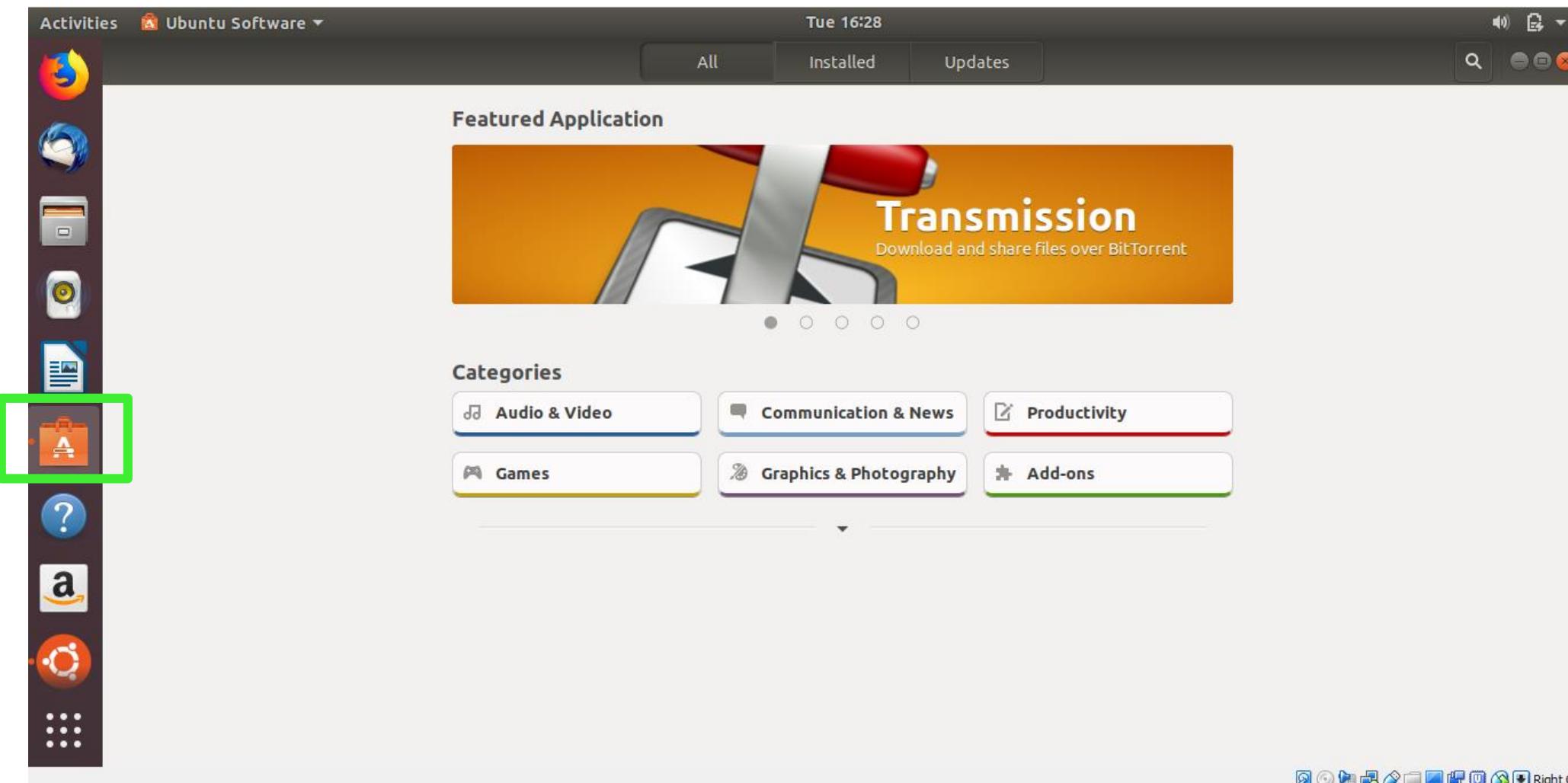
Types Of Shell



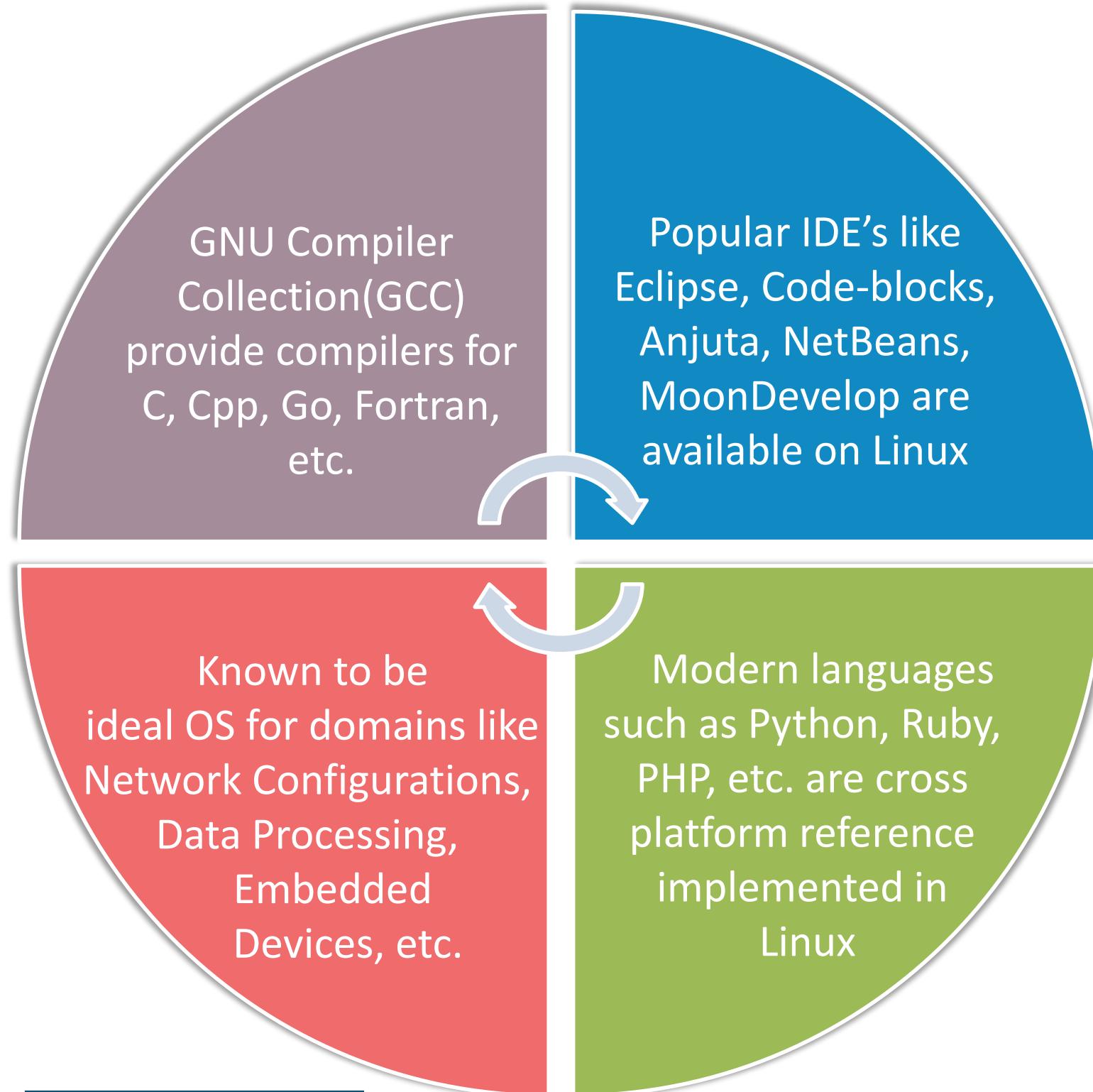
- It is a native root shell of BSD based systems
- It is essentially a C shell with programmable command line completion, command line editing and other new features
- tsch is backward compatible with original C shell
- csh is actually a Tcsh shell on many systems such as Mac OS and Red Hat Linux

Applications

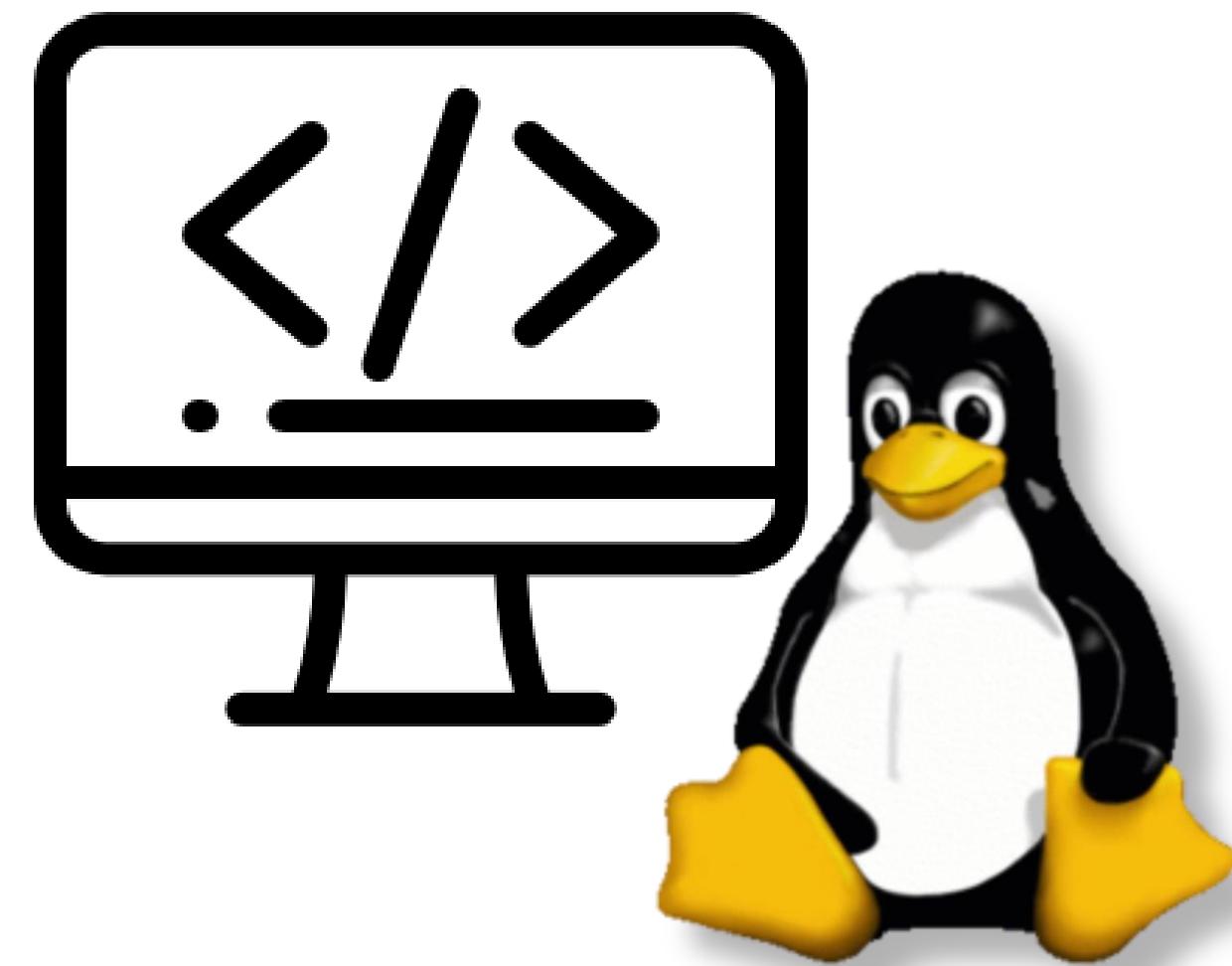
- Linux offers thousands of high-quality software titles that can be easily found and installed
- Most modern Linux distributions include App Store-like tools that centralize and simplify application installation
- System Utility programs are responsible for specialized individual level tasks

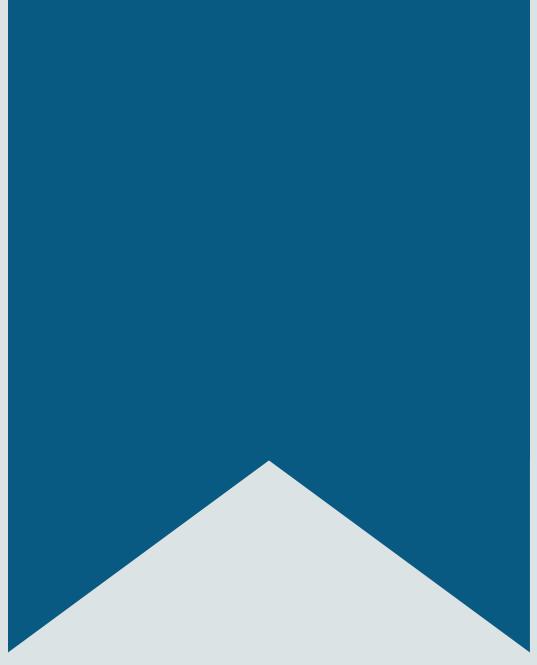


Programming In Linux



Linux Distro supports many of the programming languages

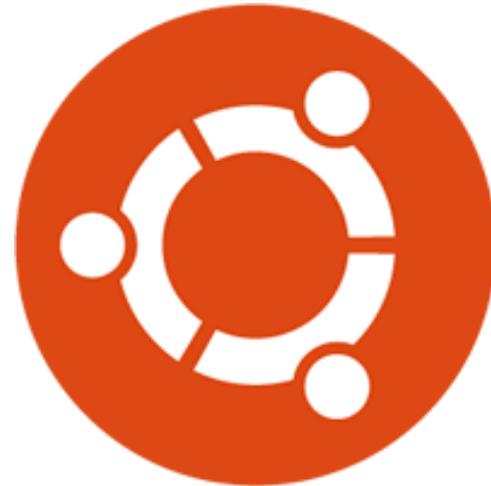




Linux Distribution

What Is Linux Distribution(Distro)?

It is an Operating System having Linux kernel and GNU tools packaged with some more applications



Ubuntu



Debian



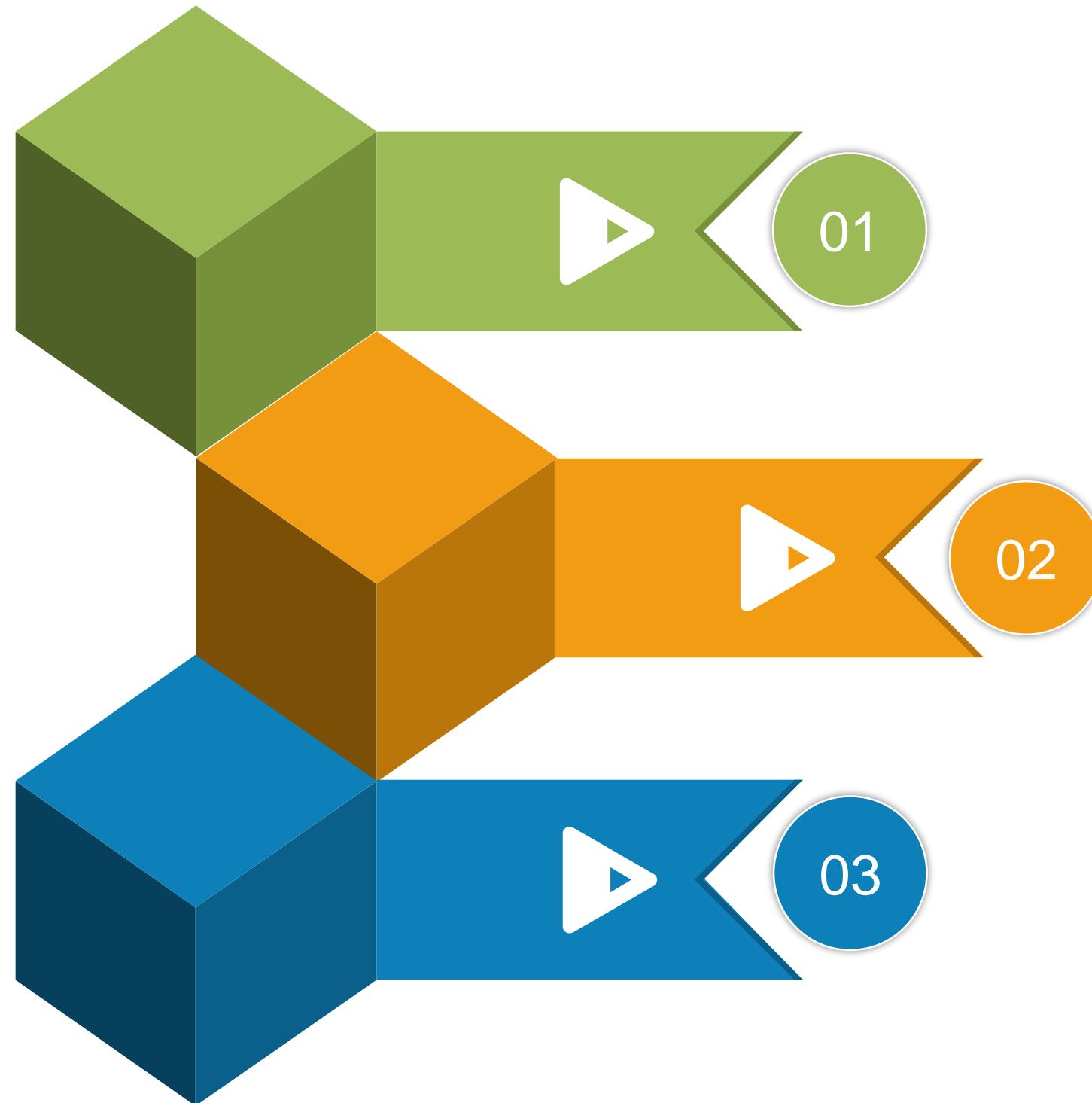
RedHat Linux



Fedora

- There are more than 600 distribution available based on:
 - The development group
 - Their specific requirement and
 - Customization
- They are community and commercial supported distros
- Some of the popular Linux Distros are Ubuntu, Fedora, RedHat, Debian, CentOS, etc.

Distinguishing Factors For Linux Distro



Enterprise users or home users

- Home user distro has user-friendly GUI and it is easy to use
- Enterprise edition gives more importance to performance

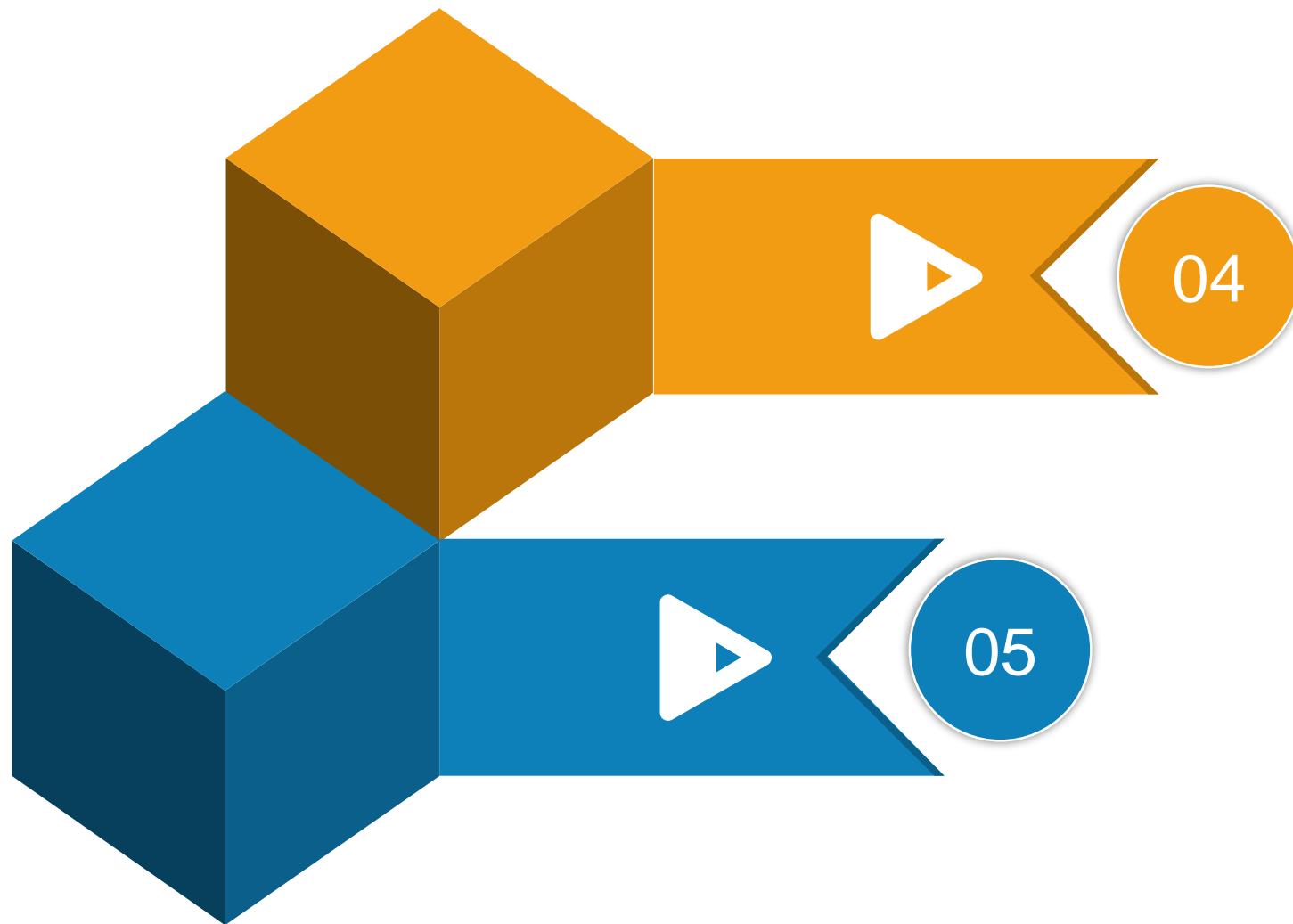
Hardware support

Most of the distros are portable to multiple hardware, but some of them are designed for specific vendors

Designed for Servers, Desktops and Embedded devices

- Server distro generally don't have a GUI
- Some of the packages of server distro are different from desktop distro

Distinguishing Factors For Linux Distro



Targeted at a specific user community

Some may have enhanced firewall securities while some may be more dedicated towards scientific computing packages

Commercial

If it is to make available commercially or non-commercially

How To Choose Distro?

The distro should be chosen based on the requirement of the user

Purpose of Use - Personal or Professional?

Easy to Install - Configuration is done with default values or it is manually chosen?

Look and Feel - Graphical Interface or Command Line interface?



For this training, it is recommended to use Ubuntu

Ubuntu



debian



- Most Popular Linux Distro
- Secure and reliable
- Multi-variant releases. Ex –Desktop, Server
- Frequent releases
- Easy to use
- It contains a wide range of software like LibreOffice, Thunderbird, etc. and also, games such as Sudoku and chess

Fedora



debian



- Latest technology having huge support base
- New releases frequently
- Easily customizable
- Can upgrade versions without reinstalling
- Stable and Reliable
- Easy to use

Debian



debian



- One of the oldest community
- Multiple hardware support
- Access to online repositories that contain over 50,000 software packages
- Customized Packages
- New release in 2-3 years
- Stable and secure

RedHat Linux



debian



- Mostly used for big Servers
- Has a maximum number of Hardware support
- Enterprise edition available with support
- Version 3.0.3 was one of the first Linux distribution
- New release in 2-3 years
- Secure and reliable

POSIX

The Portable Operating System Interface (POSIX) have applications and programs that could be moved to different computer systems without recoding

1

It is a standard set by IEEE computing society to keep compatibility between operating Systems.

2

Unix was chosen as a baseline for standardization.

3

Many user level utilities like awk, echo, were also made standard.

4

POSIX also defines a standard threading API which is supported by most OS.



Miscellaneous Linux Concepts

File

File

Run Level

Pipe



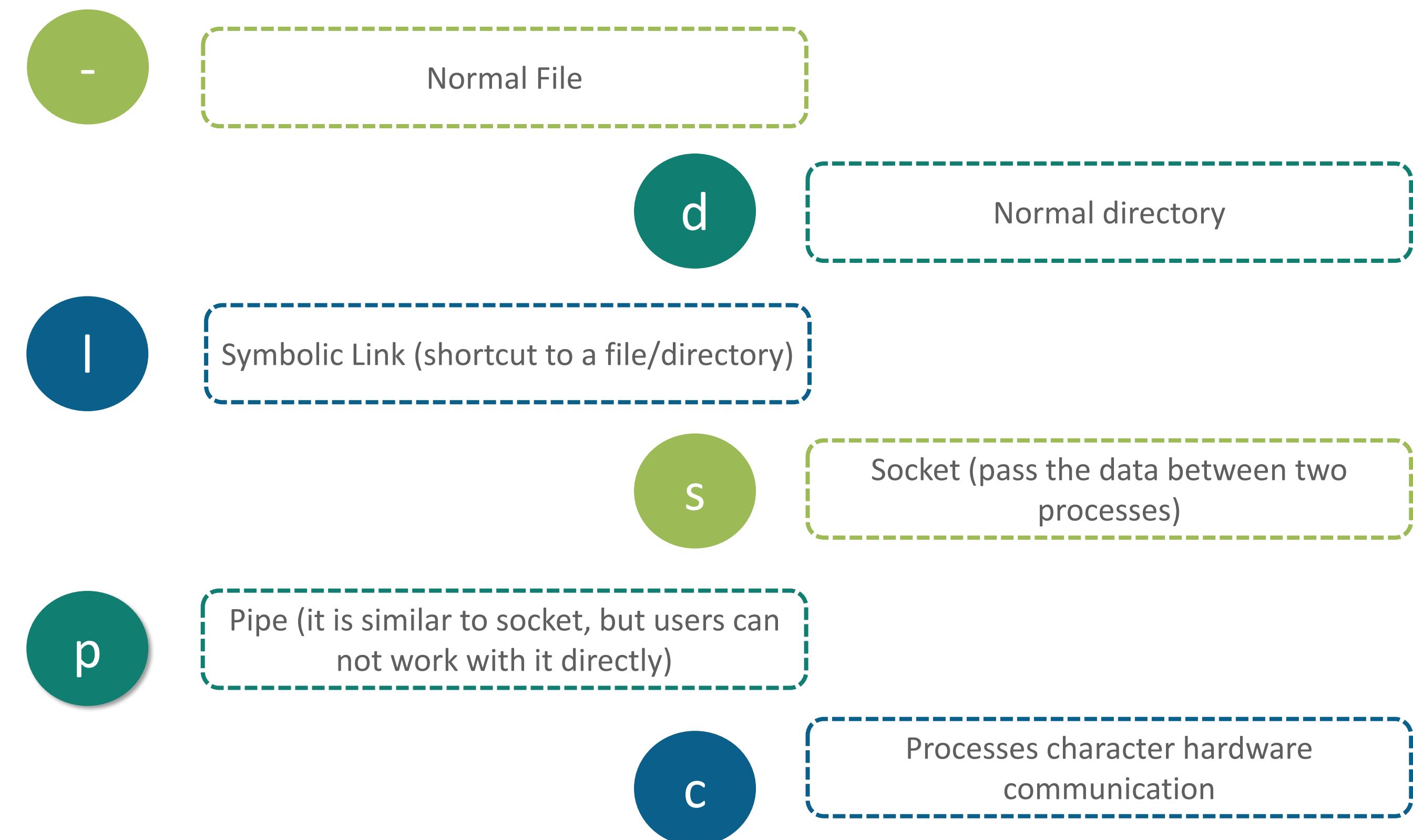
- You don't need a command to check the CPU info, but can print the file “/proc/cpuinfo” like a simple plain text file
- There are some special files that represent hardware devices, system information, etc.
- The /proc filesystem in Linux gives detail about the kernel's run-time operations in form of plain text files

Types Of File In Linux

File

Run Level

Pipe



Demo – File Types

File

Run Level

Pipe

- Check the first letter which is describing the file type
- Based on the distro, each file type is assigned with different color code

```
ubuntu@ubuntu#  
ubuntu@ubuntu#ls -ltr  
total 16  
-rw-rw-r-- 1 ubuntu ubuntu 1053 Apr 29 09:50 file.txt  
drwxr-xr-x 3 root root 4096 May 2 23:54 dir  
lrwxrwxrwx 1 root root 8 May 2 23:55 link_dir -> dir/dirl  
drwxr-xr-x 2 root root 4096 May 2 23:56 student  
-rv-r--r-- 1 root root 7 May 2 23:56 hello.txt  
ubuntu@ubuntu#
```

Run Levels

File

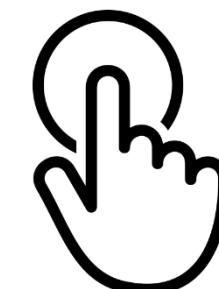
Run Level

Pipe

- The **init** figures out the default run-level to start the associated script with respect to the configured run level
- You can manually change the run level using the **telinit** command and superuser has permission to modify the run - level

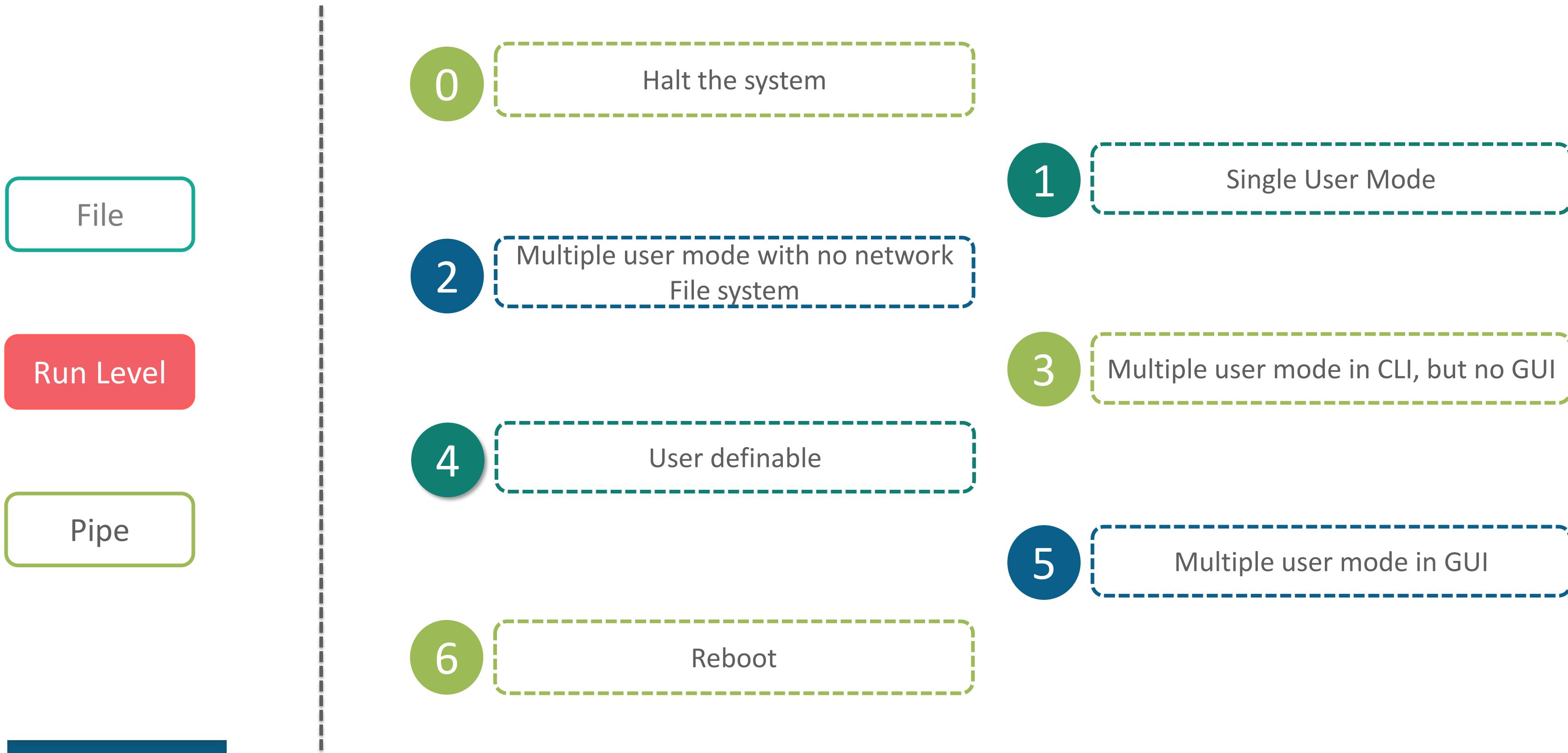


It is an operational level which describes the current state of the system with respect to the services available



It is a single digit integer that defines the state of the system

Run Level In Linux

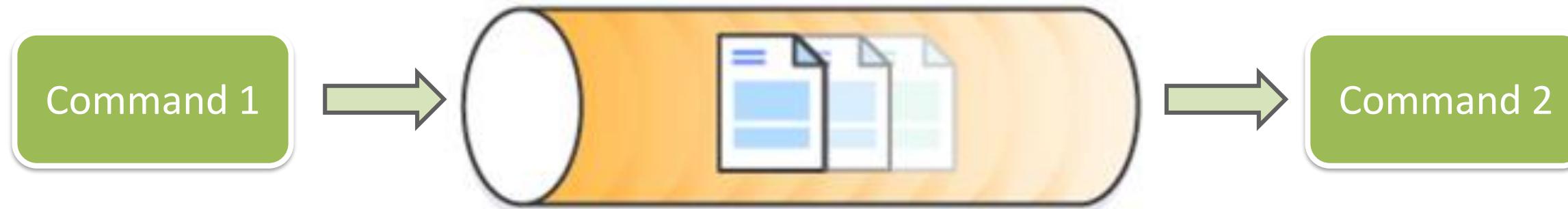


Pipe

File

Run Level

Pipe



- Pipe represented as “|” is used to direct the output of one command to another
- It creates an internal connection between two or more commands
- The data is passed to other commands directly instead of using temporary text files
- The pipe is unidirectional and data flows from left to right
- Pipe along with grep is most commonly used
- Ex – `cat fil.txt | grep "Linux"`



Installation And Shell Scripting

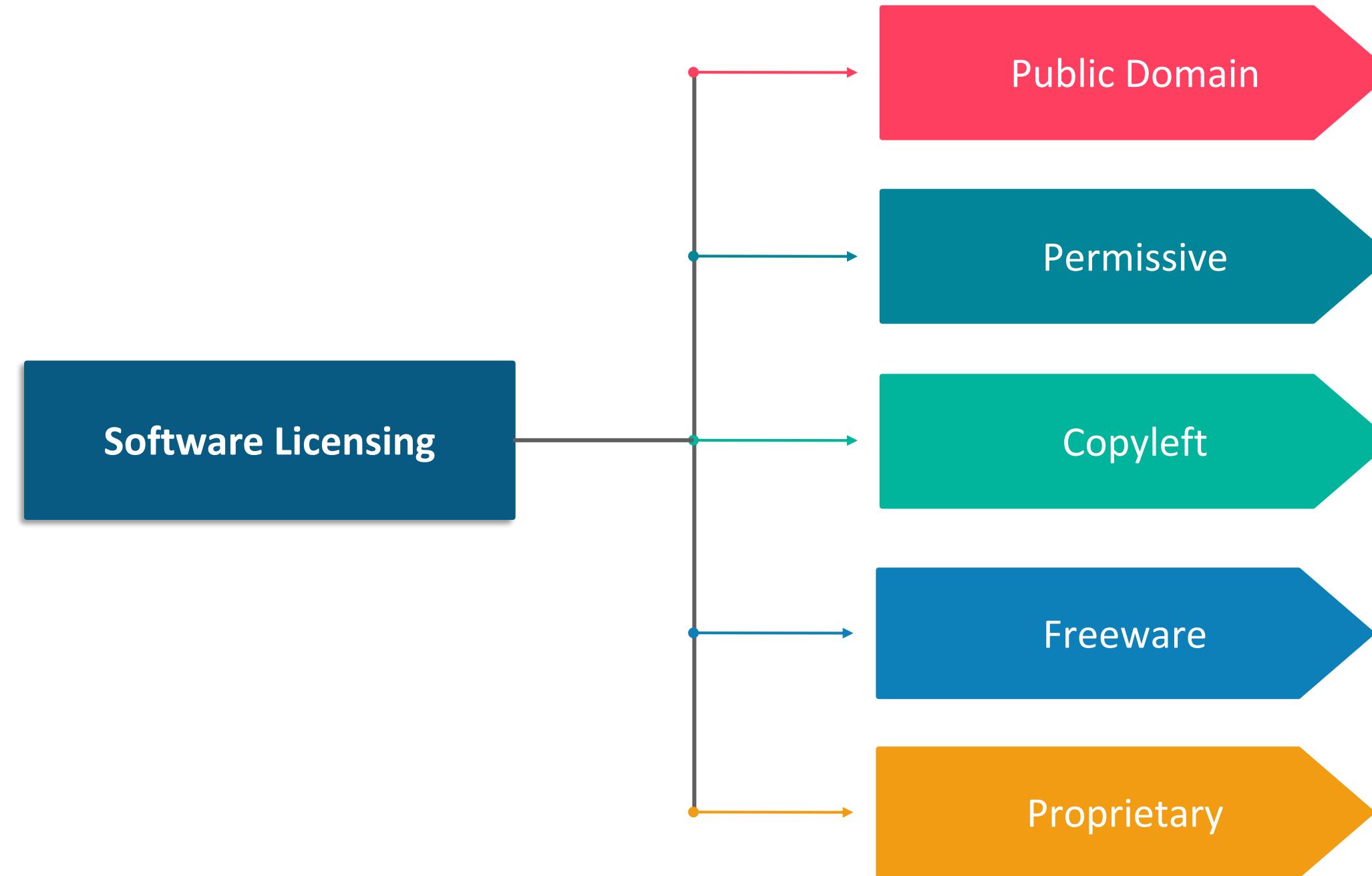


But first, let's learn
about the various
Software Licensing
types and see where
does Linux lie

What Is Software Licensing?



Types Of Software Licensing



Public Domain

- This domain has no ownership rights and can be freely used by anyone according to their requirements
- Different countries have different meanings of this license
- They may be prone to security bugs

- Allowed

- Not Allowed

Copyright Protected

Right to Copy

Right to Modify

Right to Re-distribute

Right to Sublicense

Examples – SQLite, ImageJ

Permissive

- This is the most user-friendly license
- Under same licensing, you can re-distribute the code
- You should give proper credit for the code and not make the owner liable

Copyright Protected

Right to Copy

Right to Modify

Right to Re-distribute

Right to Sublicense

- Allowed

- Not Allowed

Examples – BSD License, Apache License

Copyleft

- You need to make the source code available for re-distribution
- This is often combined with other legal obligation so it is better to read the documentation
- There may be other conditions like, it might be charged for commercial usage

Copyright Protected

Right to Copy

Right to Modify

Right to Re-distribute

Right to Sublicense

- Allowed

- Not Allowed

Examples – Linux Kernel, GIMP

Freeware

- The software is freely available for use
- You cannot modify the source code of the software
- Freeware software is often mingled with other software
- Sometimes a demo version is provided for free and you need to purchase the full version

Copyright Protected

Right to Copy

Right to Modify

Right to Re-distribute

Right to Sublincense

- Allowed

- Not Allowed

Examples – WinAmp, Acrobat Reader

Proprietary

- This maintains exclusive rights of the software as well as the owner
- The owner can restrict the use and make regular inspection
- Sharing of the software is generally restricted
- It can not override the copyright law

Copyright Protected

Right to Copy

Right to Modify

Right to Re-distribute

Right to Sublicense

- Allowed

- Not Allowed

Examples – MS Windows, Adobe Photoshop

GPL And Its Versions

1

General Public License (GPL) is free, copyleft and mandatory license for Software and other kinds of work

2

The license was originally written by Stallman for the GNU Project

3

Based on increasing complexities the licensing has undergone changes and currently is in 3rd revision

4

Version was released in 1989, which was undergone moderations. In 1991 version 2.0 was released

5

And after further modifications GPLv3 was released in 2007

6

Some major changes in GPLv3 are DRM software cannot be created with GPL license. Patents and related queries were addressed



Requirements

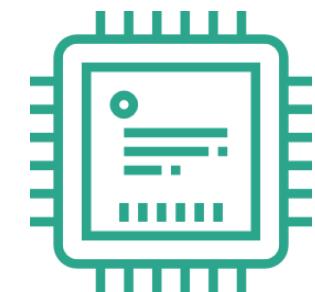
The recommended system requirements for this course are:



HARD DISK: 20GB



RAM: 2GB



POWER SOURCE: Keep connected to a power source



BOOT DEVICE: Access to CD Drive or USB

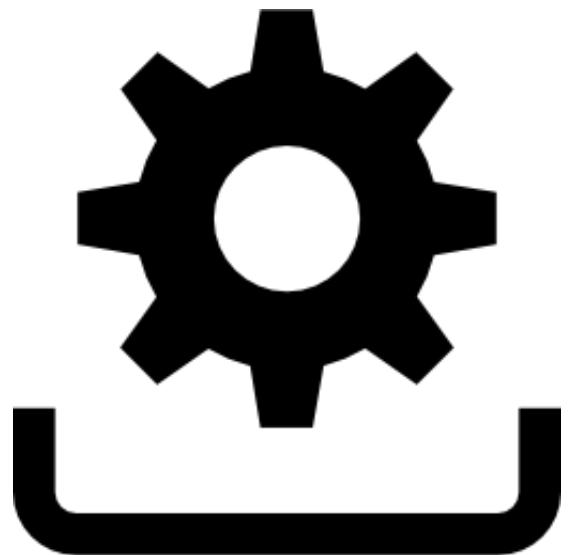
BACKUP: Maintain a backup of your current system



Installation And Initialization – Key Activities

Some of the basic steps performed during installation are:

- Making sure the fulfillment of necessary system requirements
- Checking for existing versions of the software
- Creating and updating program files and folders
- Adding configuration data such as Files, Windows Register Entries, and Environment Variables
- Making the software accessible to the user for instance by creating Links, Shortcuts, and Bookmarks
- Configuring components that run automatically, such as Daemons or Window's services
- Performing product activation
- Updating the software versions



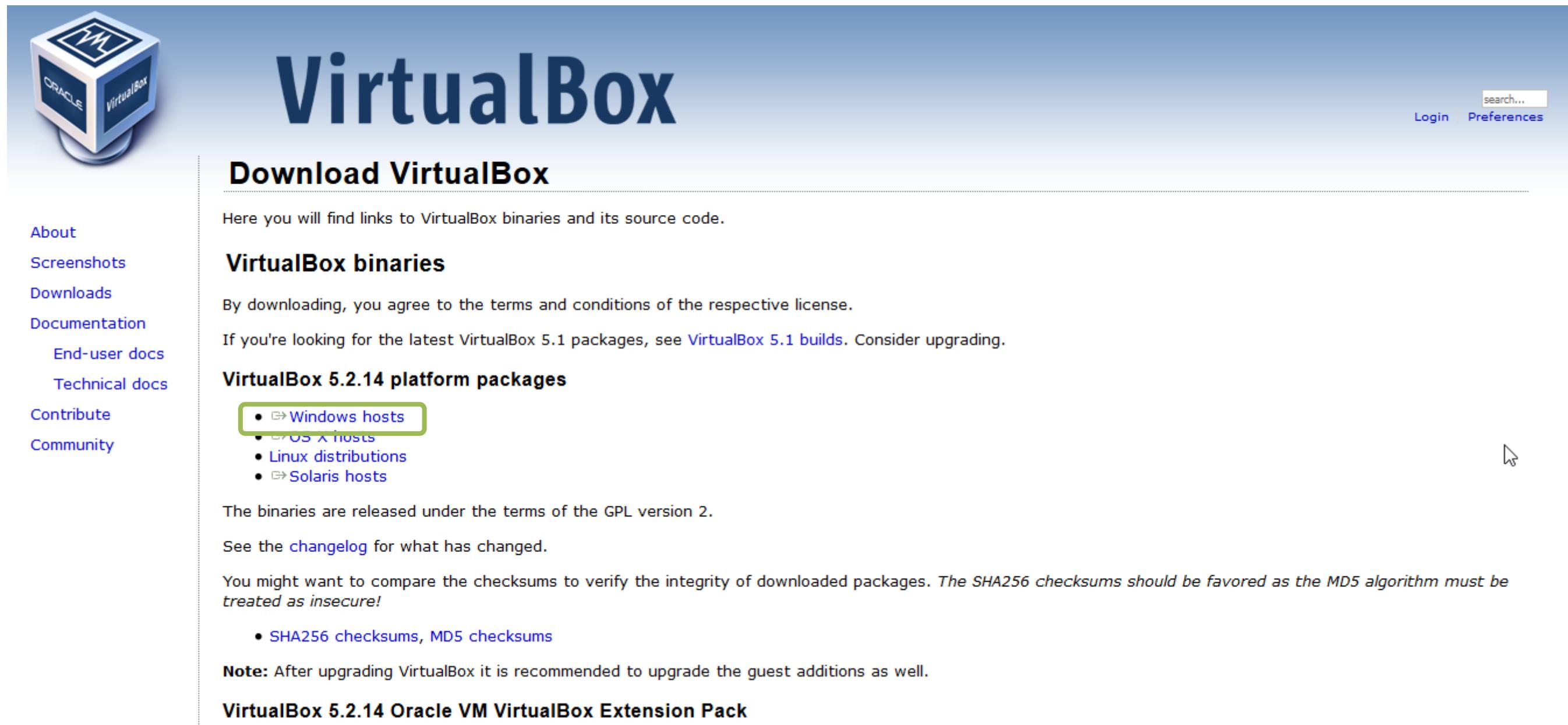
Installation And Initialization

Step 1:

- Install **VirtualBox** from given link:

[https://www.virtualbox.org
/wiki/Downloads](https://www.virtualbox.org/wiki/Downloads)

- And select windows host as we are using windows OS. For other OSes, you may select respectively.

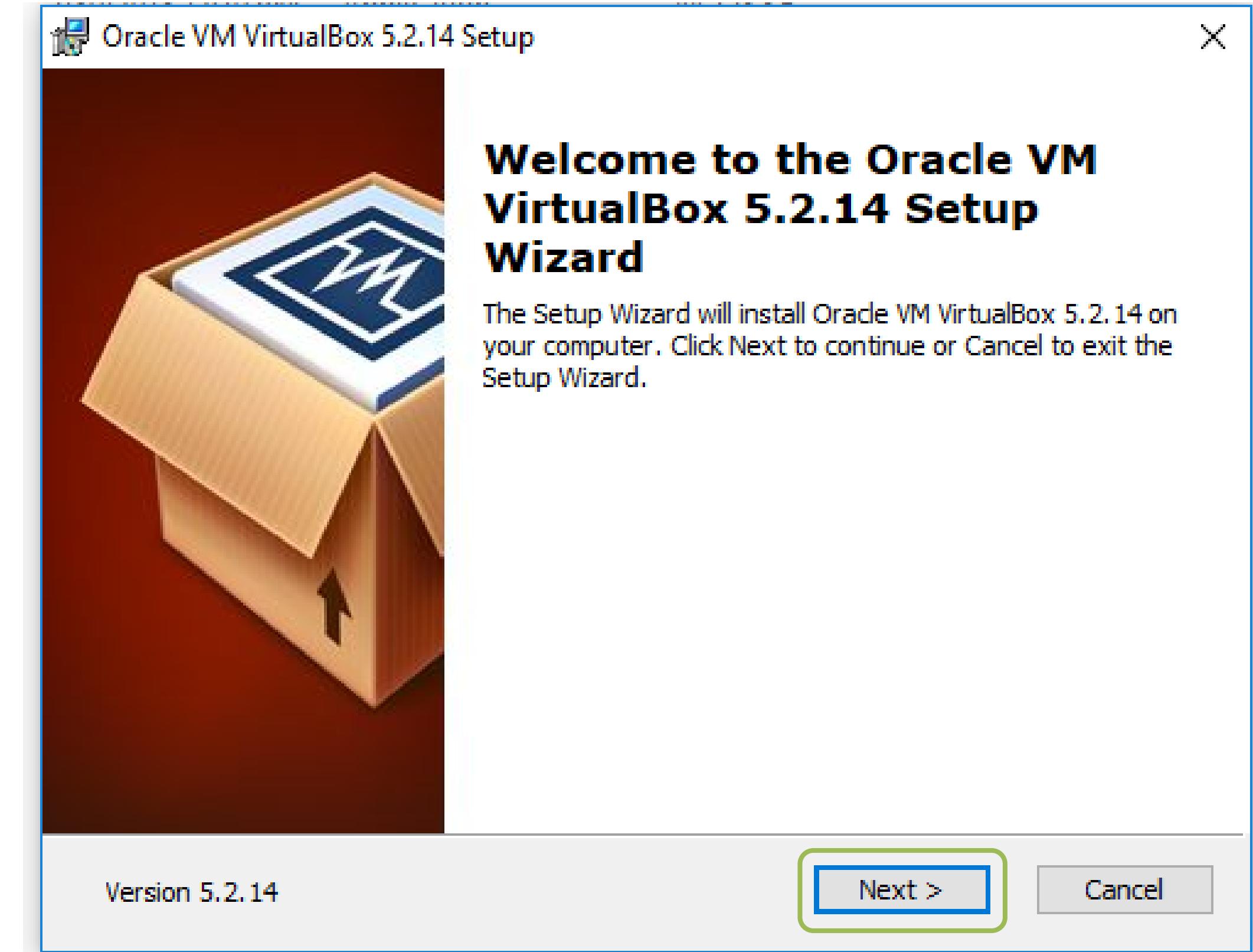


The screenshot shows the 'Download VirtualBox' page of the VirtualBox website. At the top, there is a navigation bar with links for 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. On the right side of the header, there are 'Login' and 'Preferences' buttons, as well as a search bar. The main content area features a large 'VirtualBox' logo and a sub-section titled 'Download VirtualBox'. Below this, a text block says 'Here you will find links to VirtualBox binaries and its source code.' A section titled 'VirtualBox binaries' contains a note: 'By downloading, you agree to the terms and conditions of the respective license.' It also mentions that for the latest packages, see 'VirtualBox 5.1 builds'. A section for 'VirtualBox 5.2.14 platform packages' lists download links for 'Windows hosts' (which is highlighted with a green box and a cursor), 'OS X hosts', 'Linux distributions', and 'Solaris hosts'. A note below states that binaries are released under the terms of the GPL version 2. It also links to the 'changelog' for changes. A note about checksums says: 'You might want to compare the checksums to verify the integrity of downloaded packages. The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!' It lists 'SHA256 checksums, MD5 checksums'. A note at the bottom says: 'Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.' Finally, a link for the 'VirtualBox 5.2.14 Oracle VM VirtualBox Extension Pack' is provided.

Installation And Initialization

Step 2:

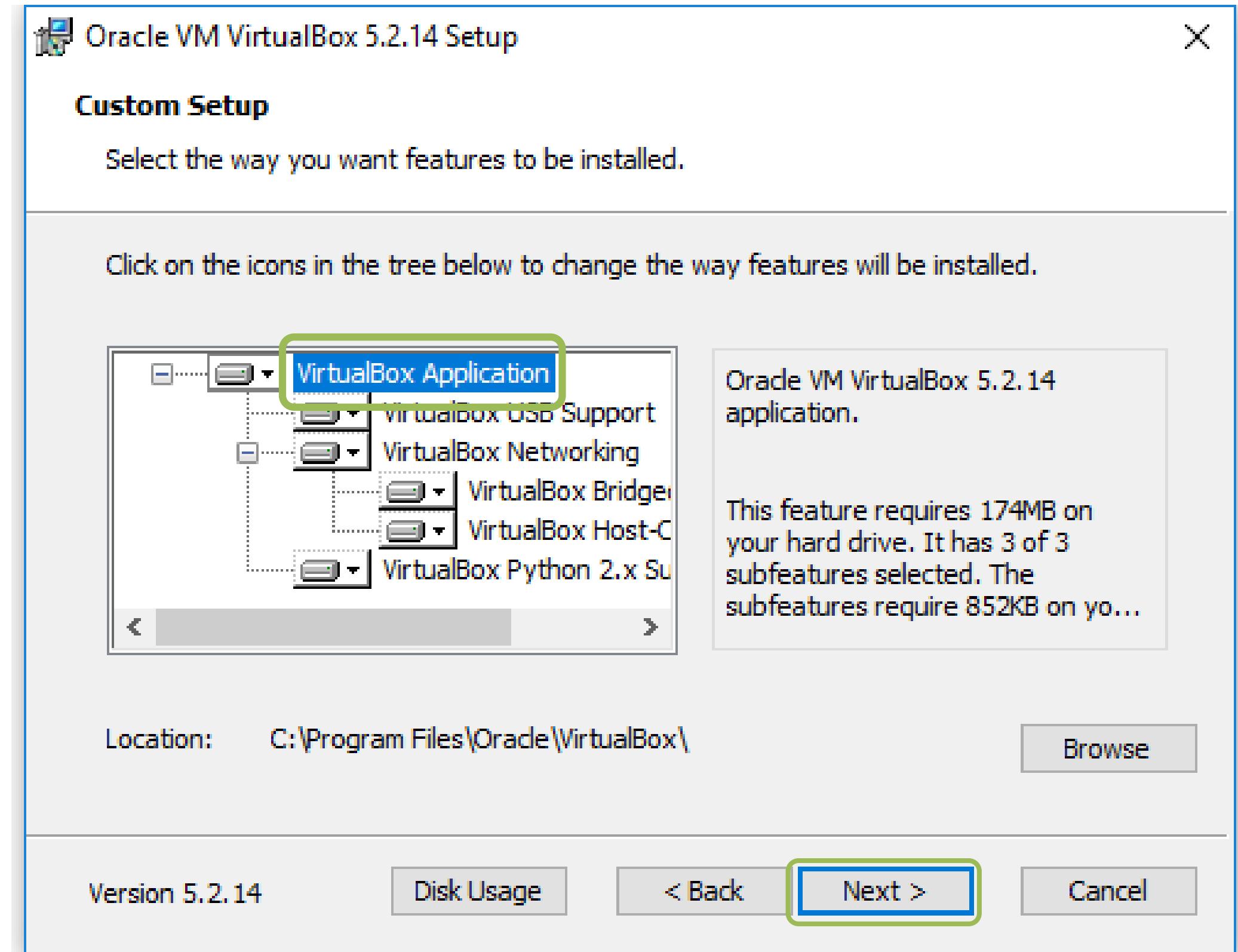
Once it is downloaded, Open the File, you will get this window and click on Next button.



Installation And Initialization

Step 3:

Click on **VirtualBox** Application and then click on **Next**.

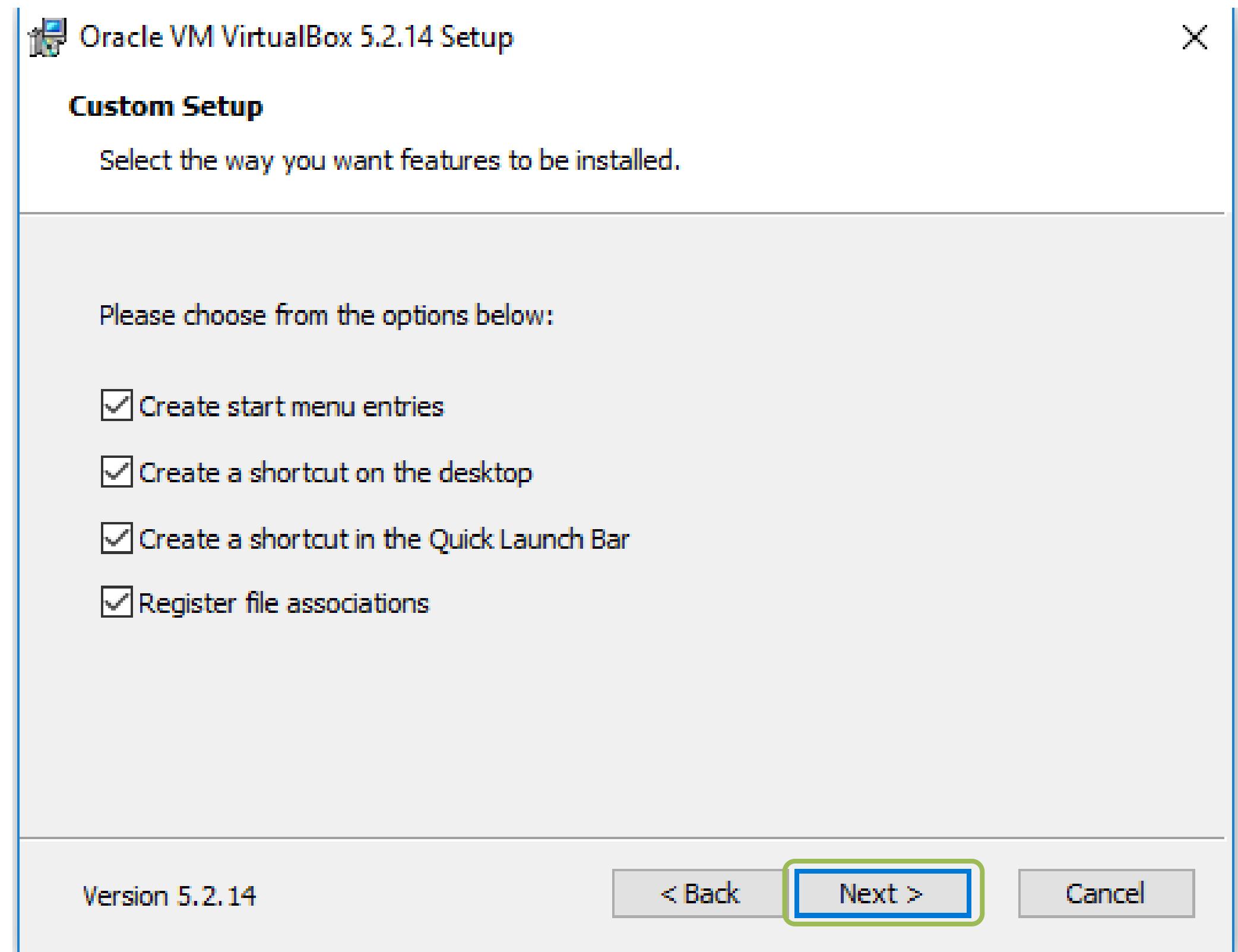


Installation And Initialization

Step 4:

By default Features will be selected.

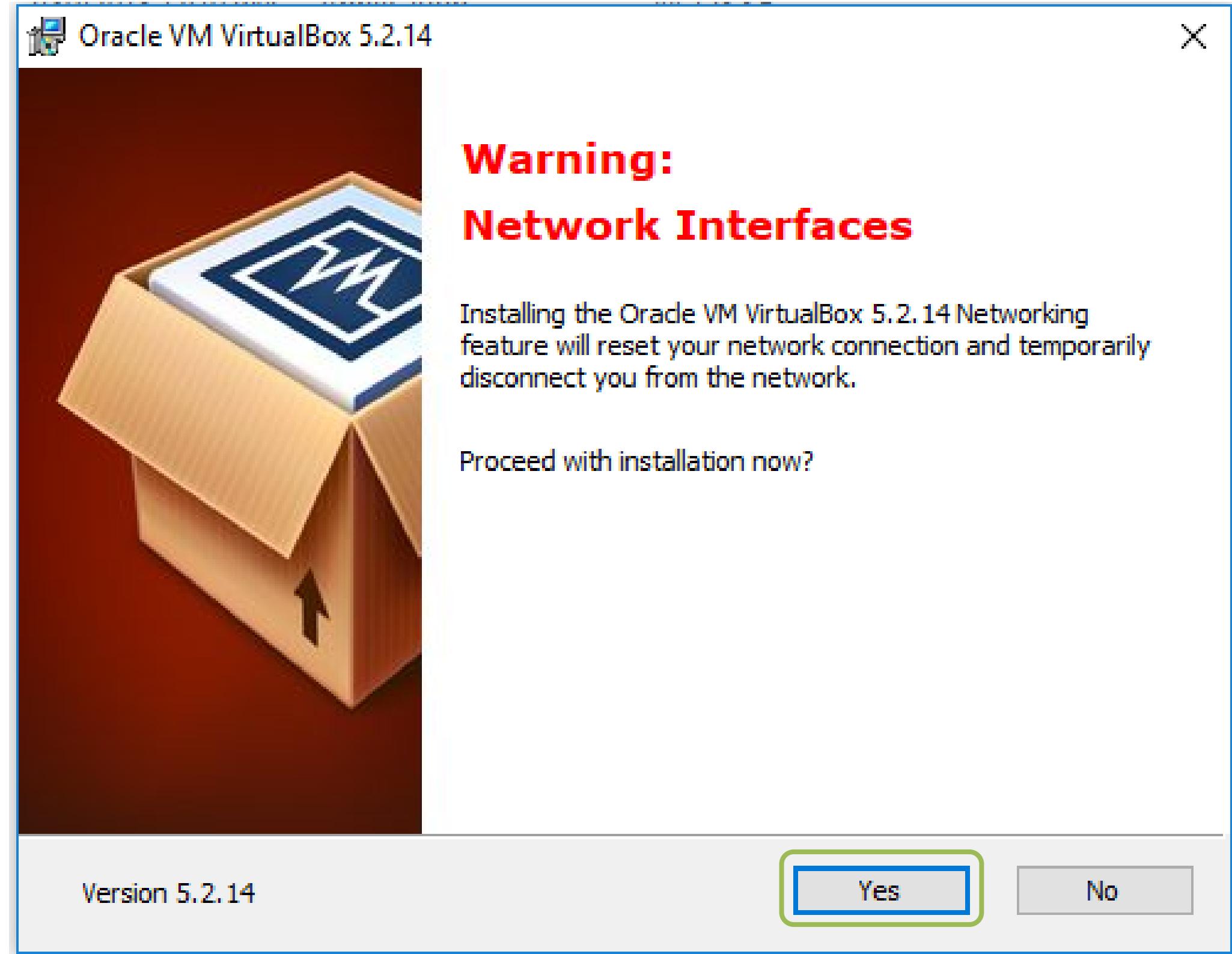
Click on **Next** button.



Installation And Initialization

Step 5:

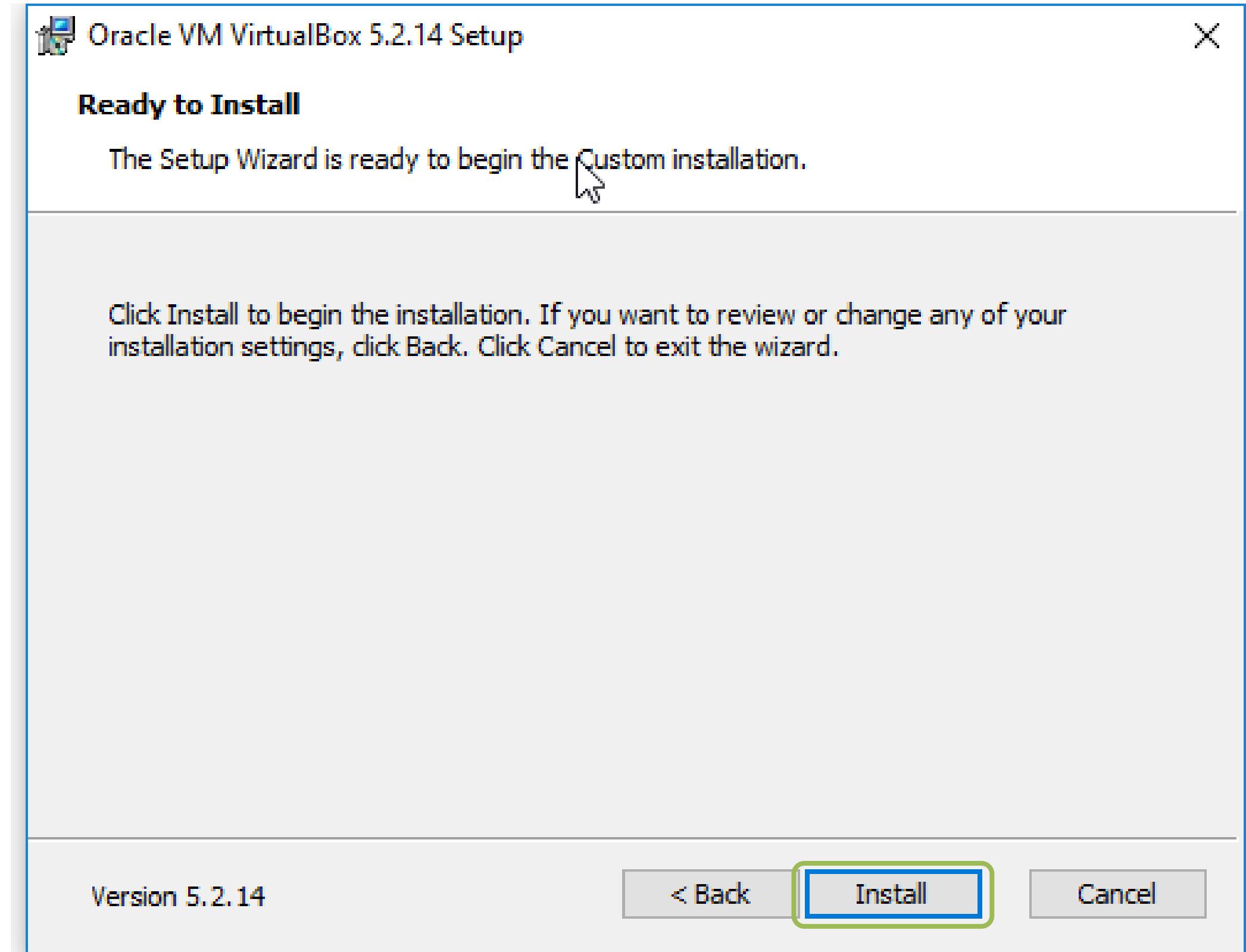
Here the warning will be popped for network interfaces, you need not worry. All the issues will be fixed once VirtualBox is installed. Click on **Yes**.



Installation And Initialization

Step 6:

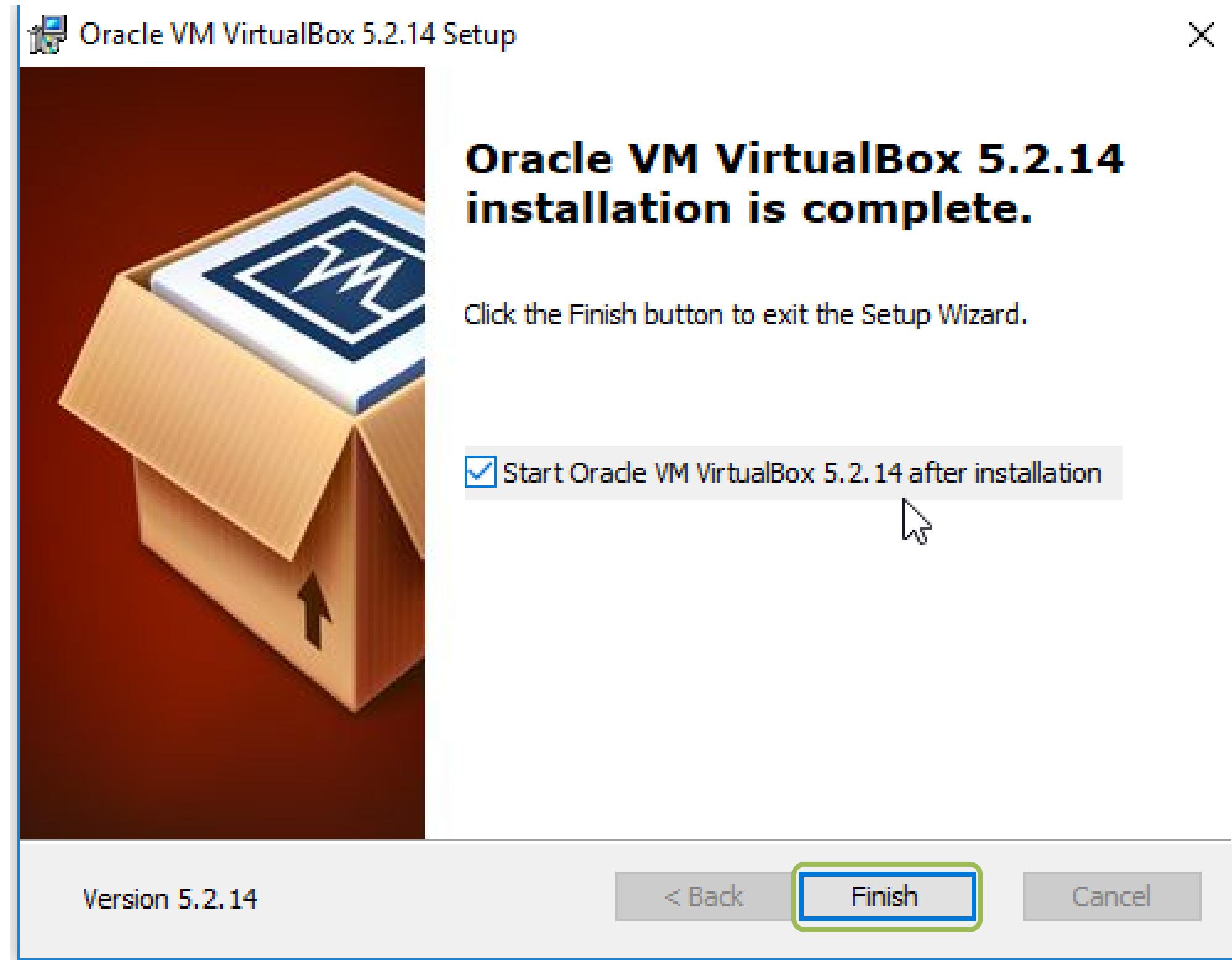
Click on **Install**.



Installation And Initialization

Step 7:

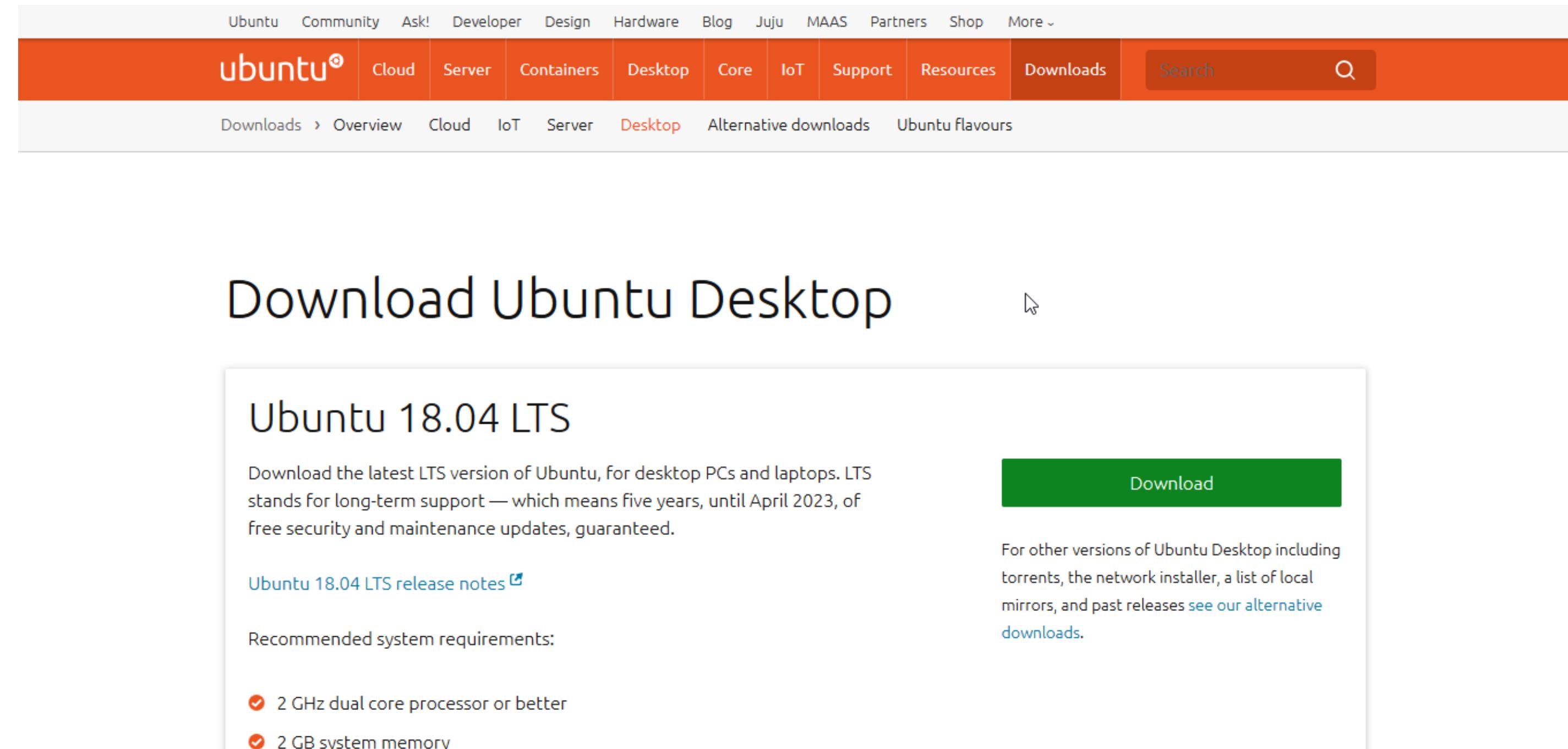
Click on **Finish**, your VirtualBox will be installed and you may see its shortcut on your desktop.



Installation And Initialization

Step 8:

Download **Ubuntu 18.04 LTS**. Link is given here:
<https://www.ubuntu.com/download/desktop>



The screenshot shows the Ubuntu website's download section. At the top, there's a navigation bar with links like Ubuntu, Community, Ask!, Developer, Design, Hardware, Blog, Juju, MAAS, Partners, Shop, and More. Below the navigation is a secondary navigation bar with links for Cloud, Server, Containers, Desktop, Core, IoT, Support, Resources, and Downloads. The 'Desktop' link is highlighted in red. The main content area has a large heading 'Download Ubuntu Desktop' with a cursor icon pointing towards it. Below the heading is a section for 'Ubuntu 18.04 LTS' with a 'Download' button. To the right, there's a link to 'Alternative downloads'.

Ubuntu 18.04 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2023, of free security and maintenance updates, guaranteed.

[Ubuntu 18.04 LTS release notes](#)

Recommended system requirements:

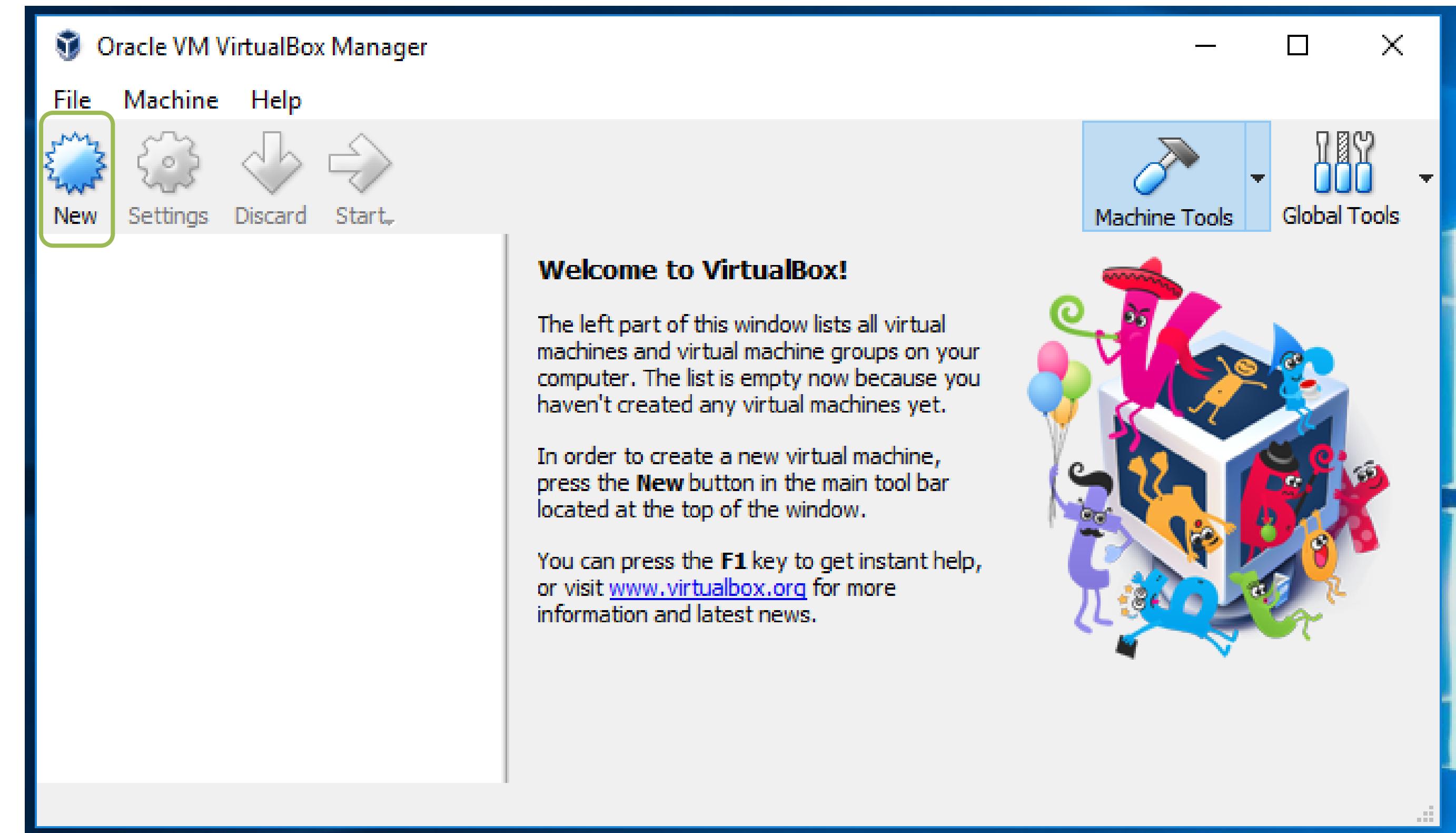
- 2 GHz dual core processor or better
- 2 GB system memory

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases [see our alternative downloads](#).

Installation And Initialization

Step 9:

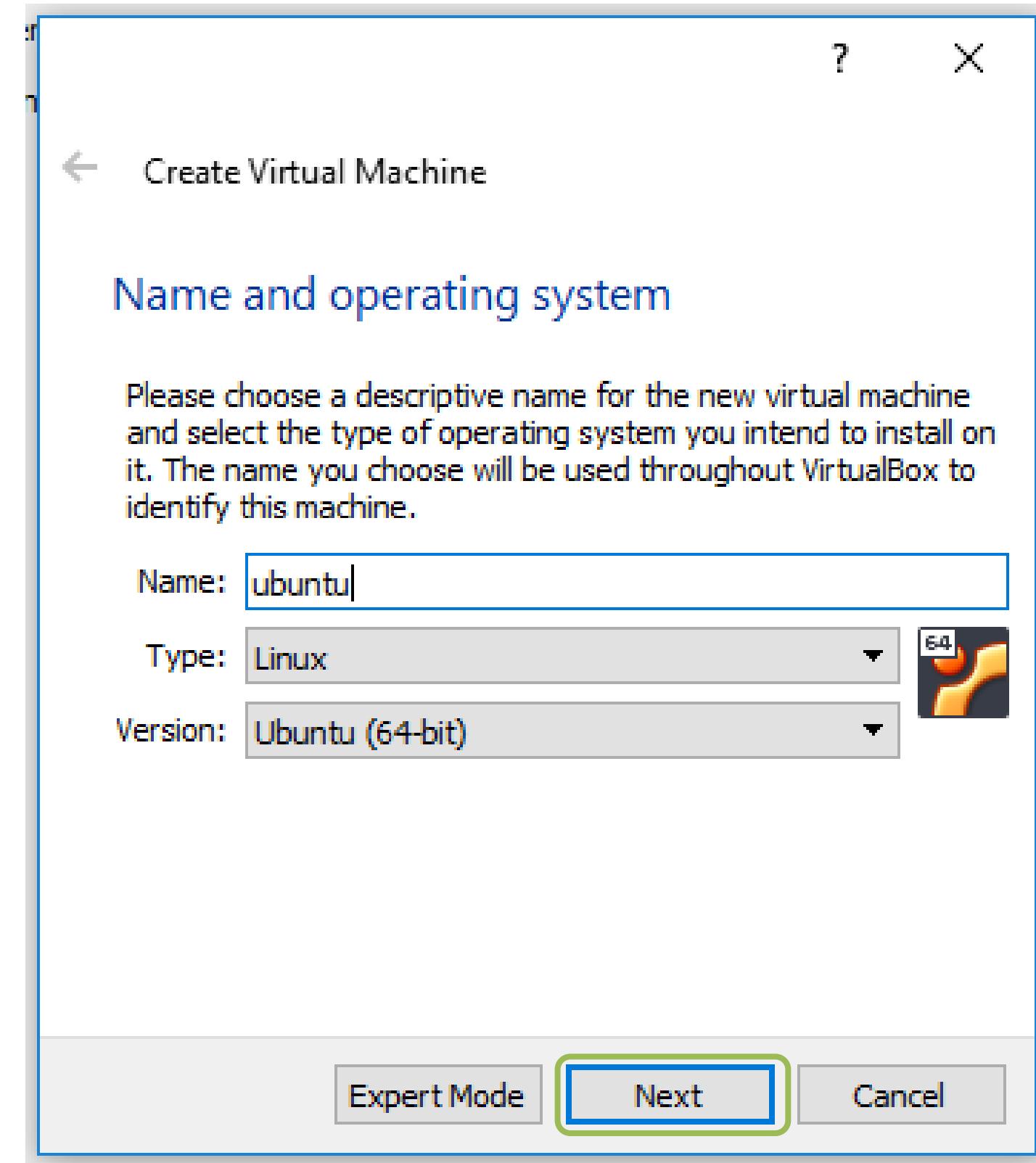
Open VirtualBox and click on **New** on the top right corner.



Installation And Initialization

Step 10:

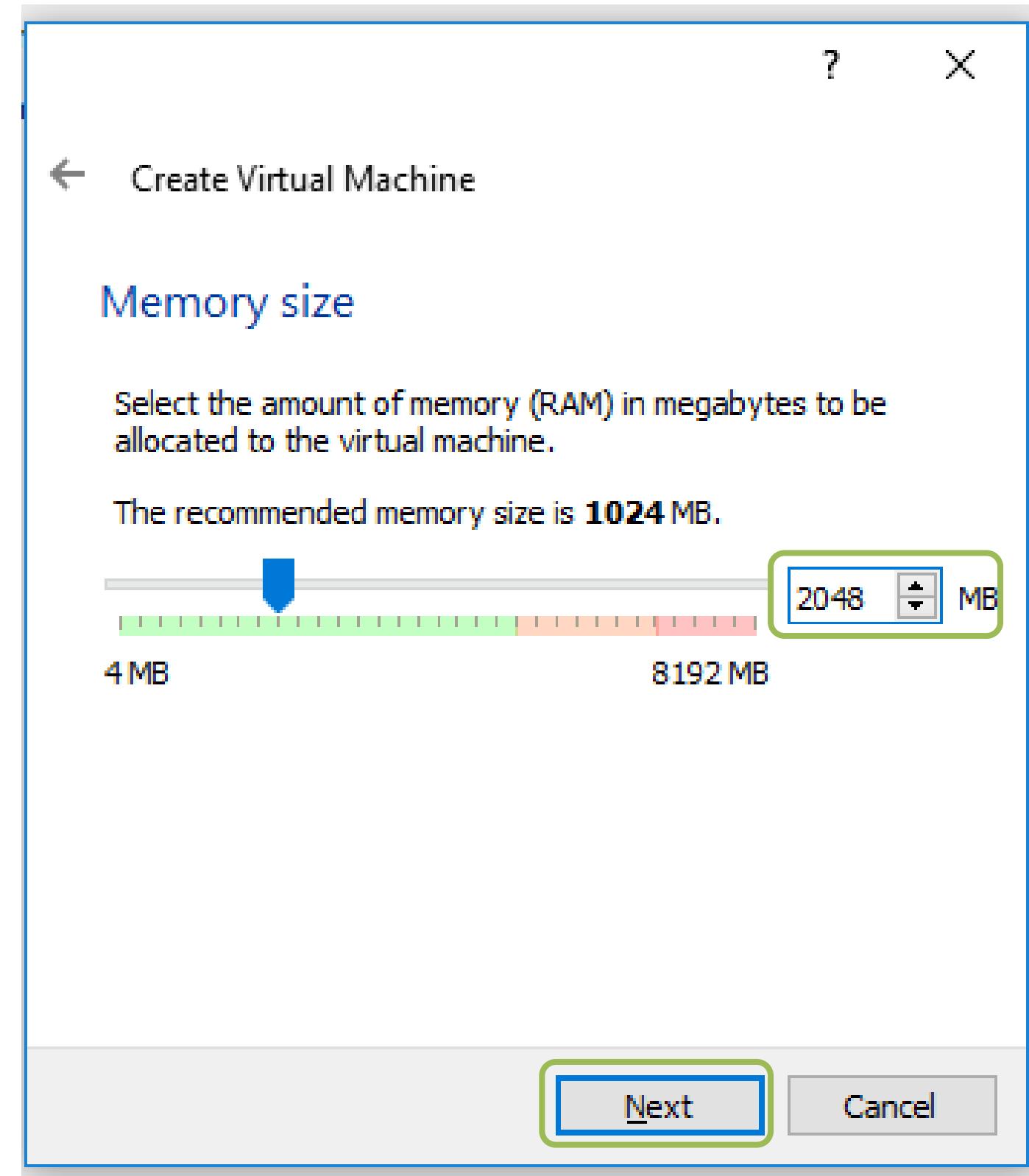
In front of Name type **Ubuntu**. It automatically selects the type and version.
Click on **Next**.



Installation And Initialization

Step 11:

Select minimum RAM as 2048MB and click **Next**.

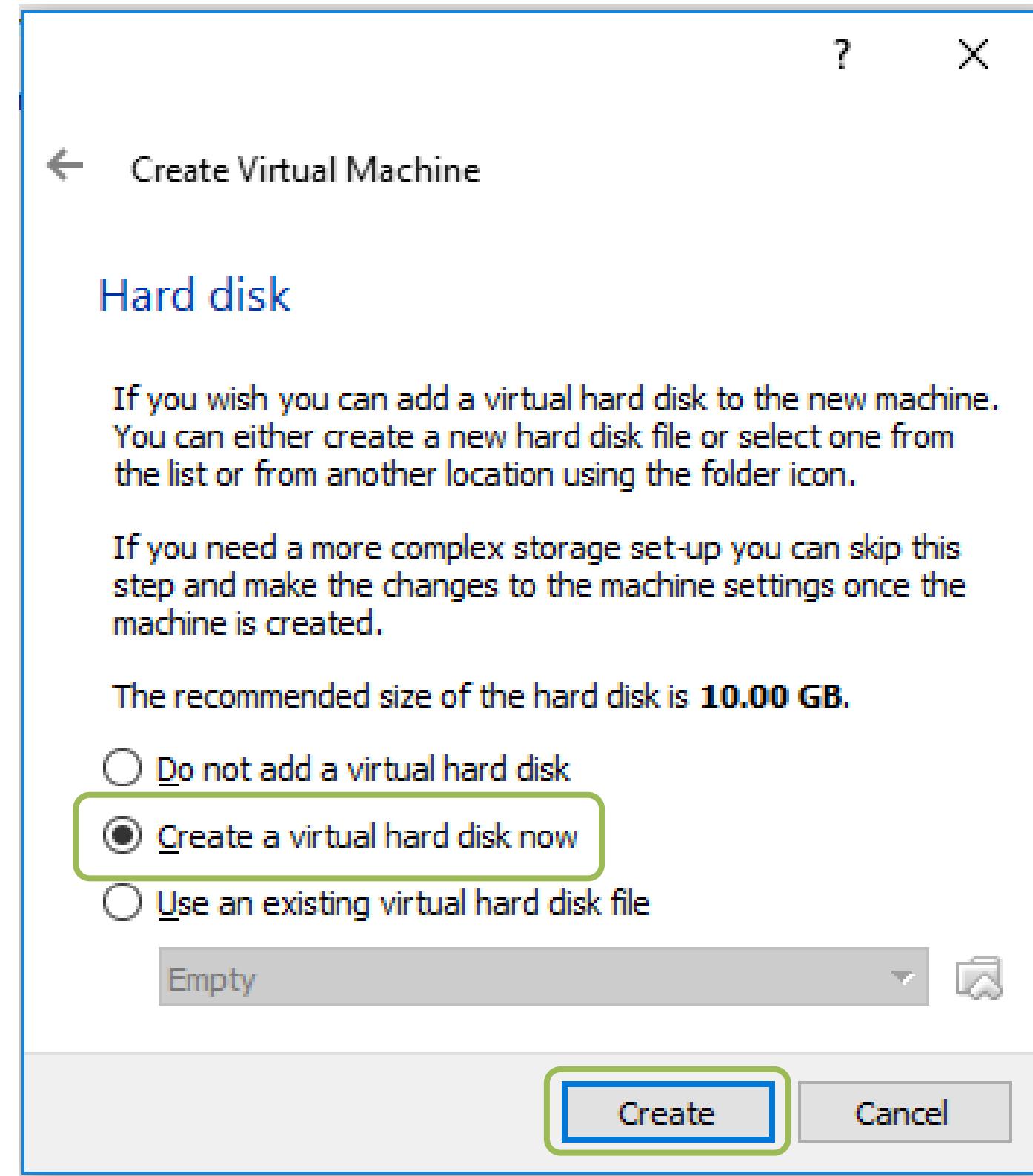


Installation And Initialization

Step 12:

Select 2nd option i.e. Create a Virtual hard disk now.

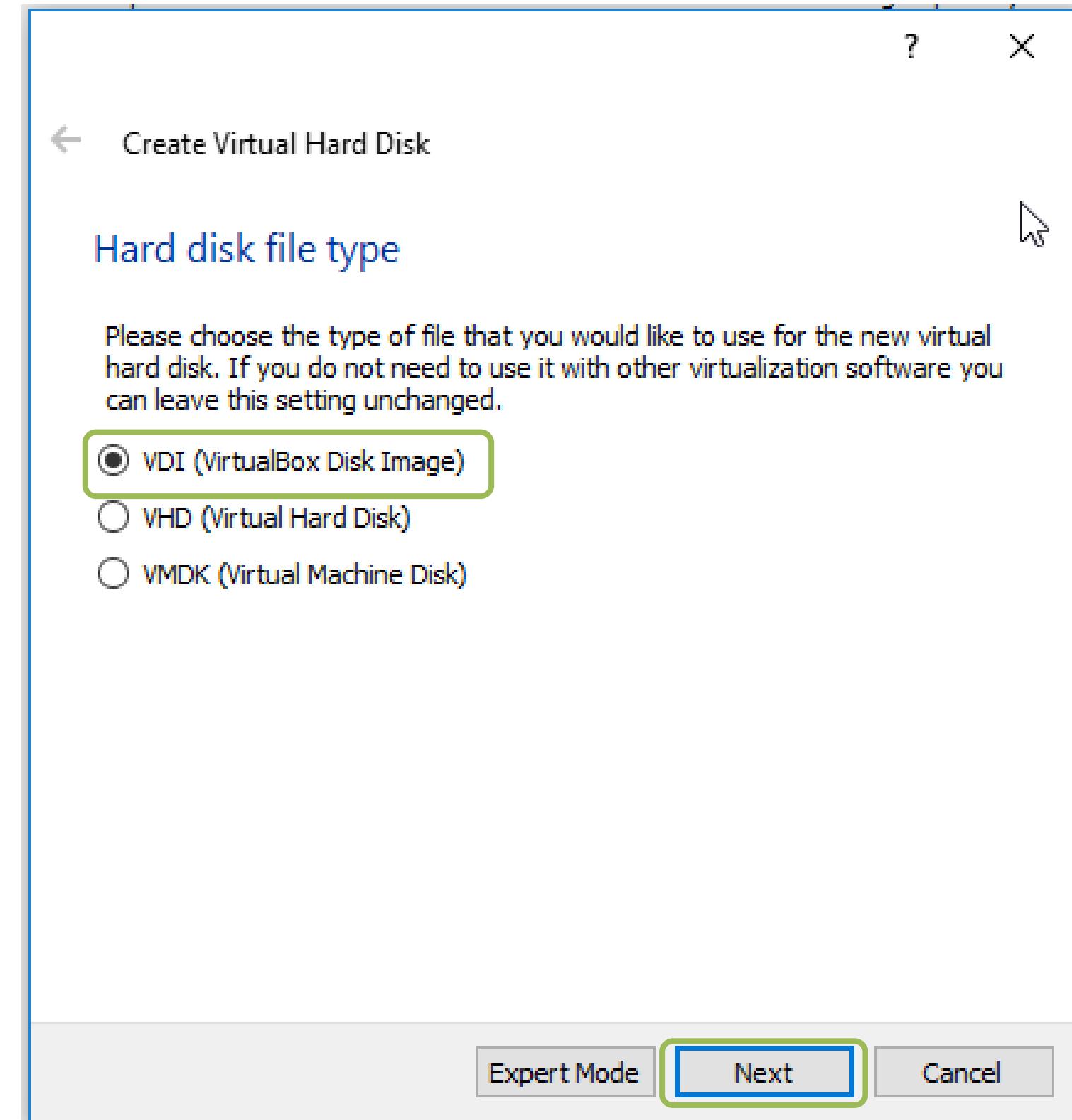
Click on **Create**.



Installation And Initialization

Step 13:

Select VDI (VirtualBox Disk Image).
Click on Next.

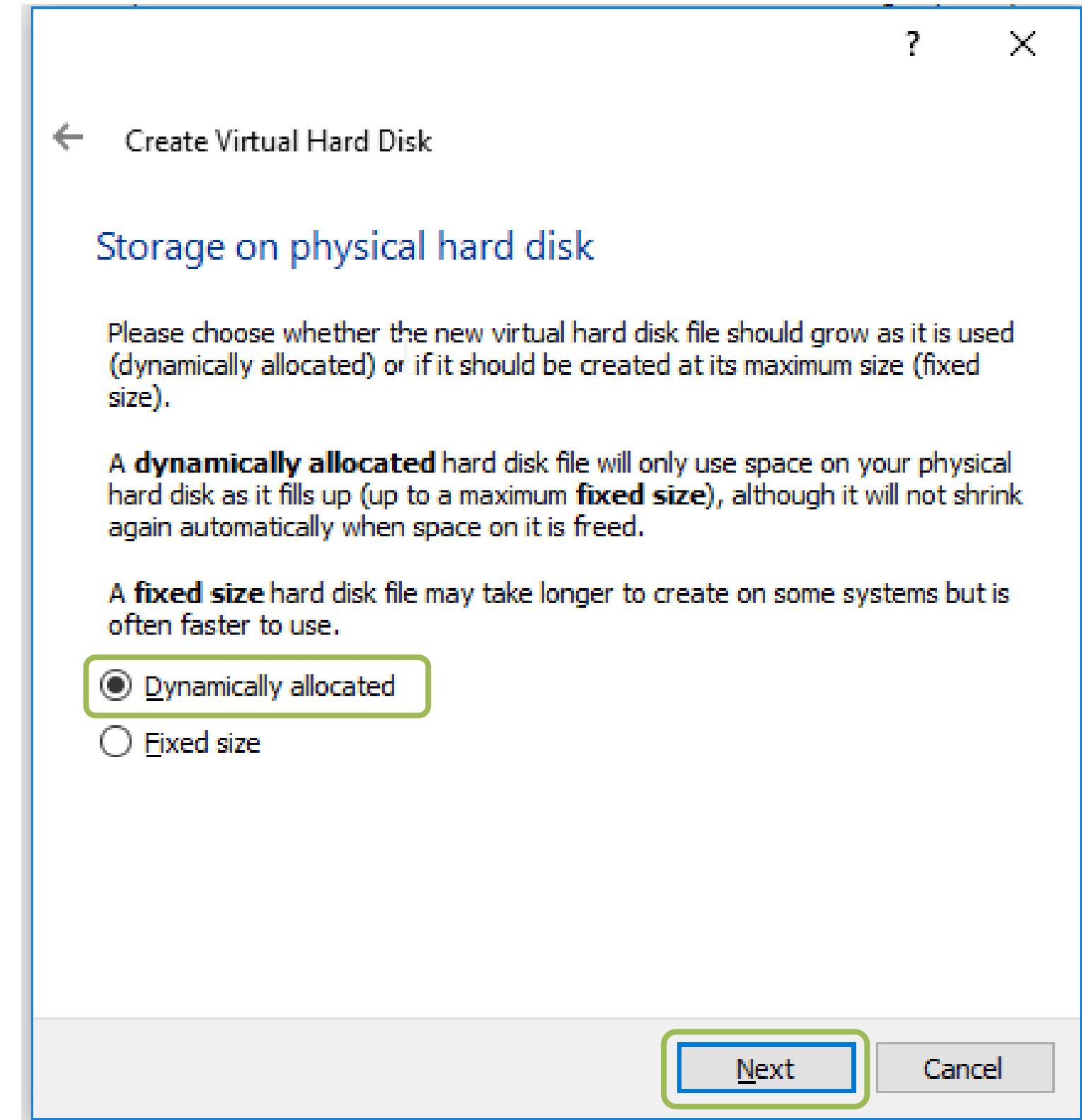


Installation And Initialization

Step 14:

For the storage on Hard Disk, select **Dynamically allocated**.

Click on **Next**.

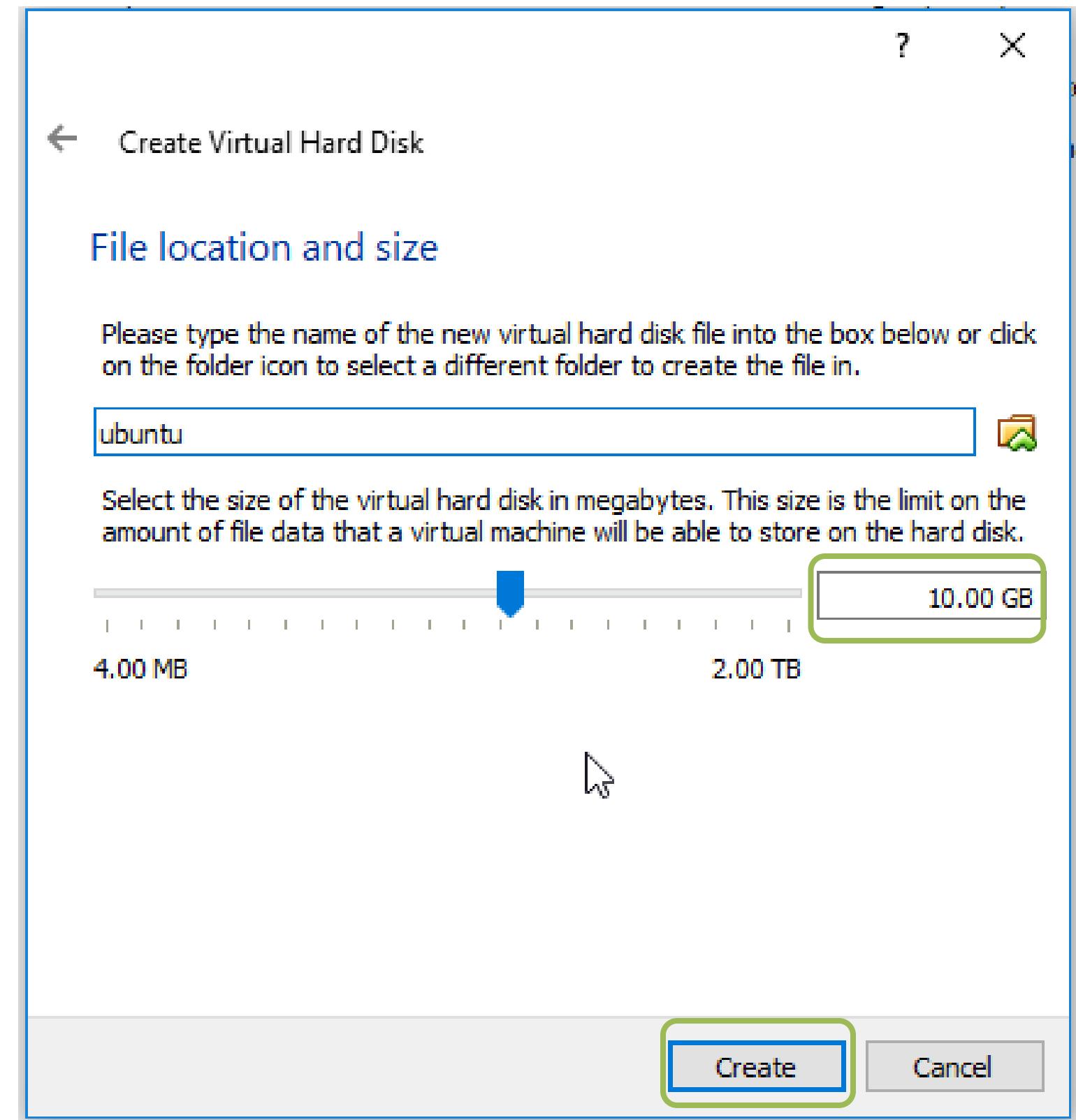


Installation And Initialization

Step 15:

It is recommended to allocate 10GB of HDD space.

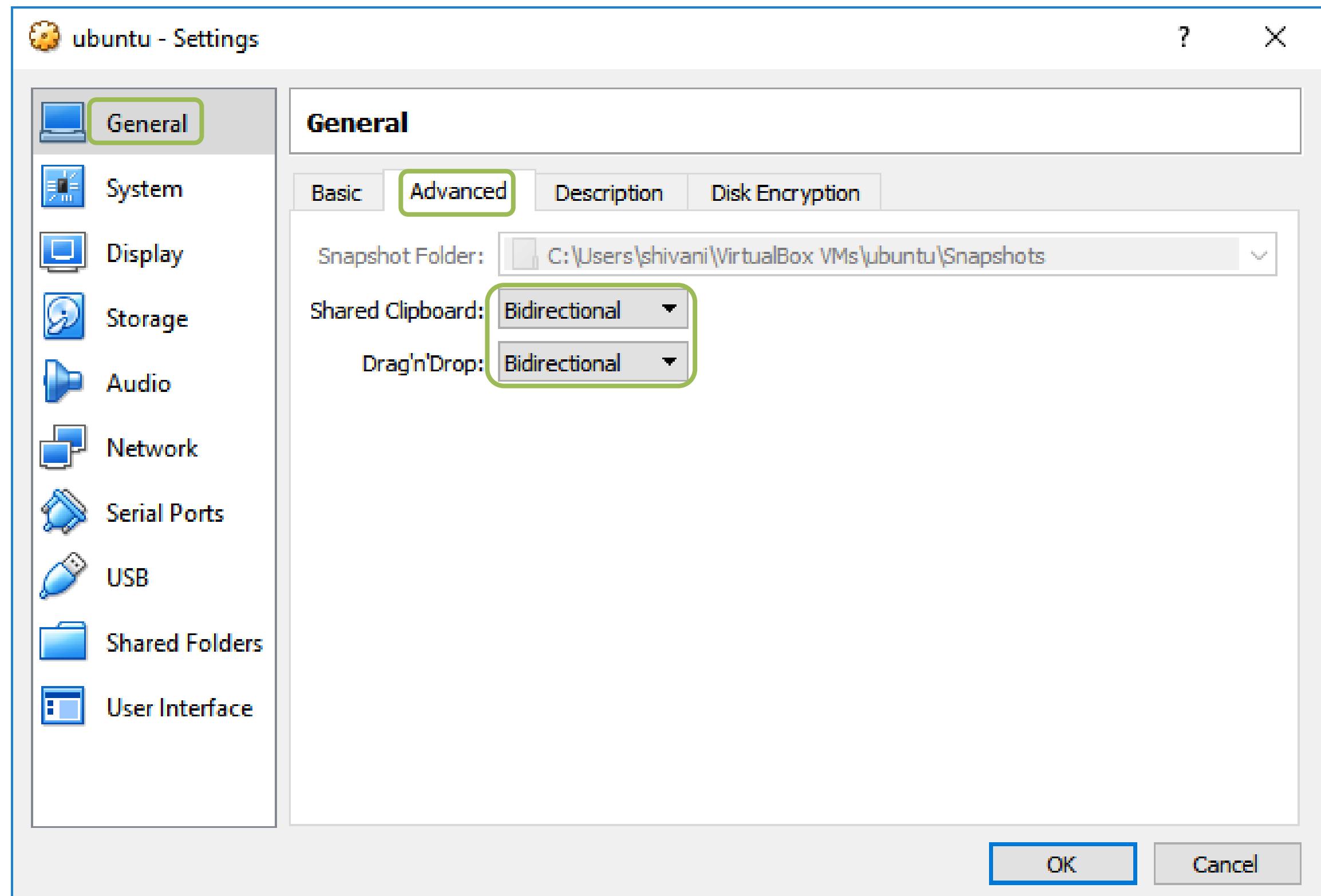
Click on **Create**.



Installation And Initialization

Step 16:

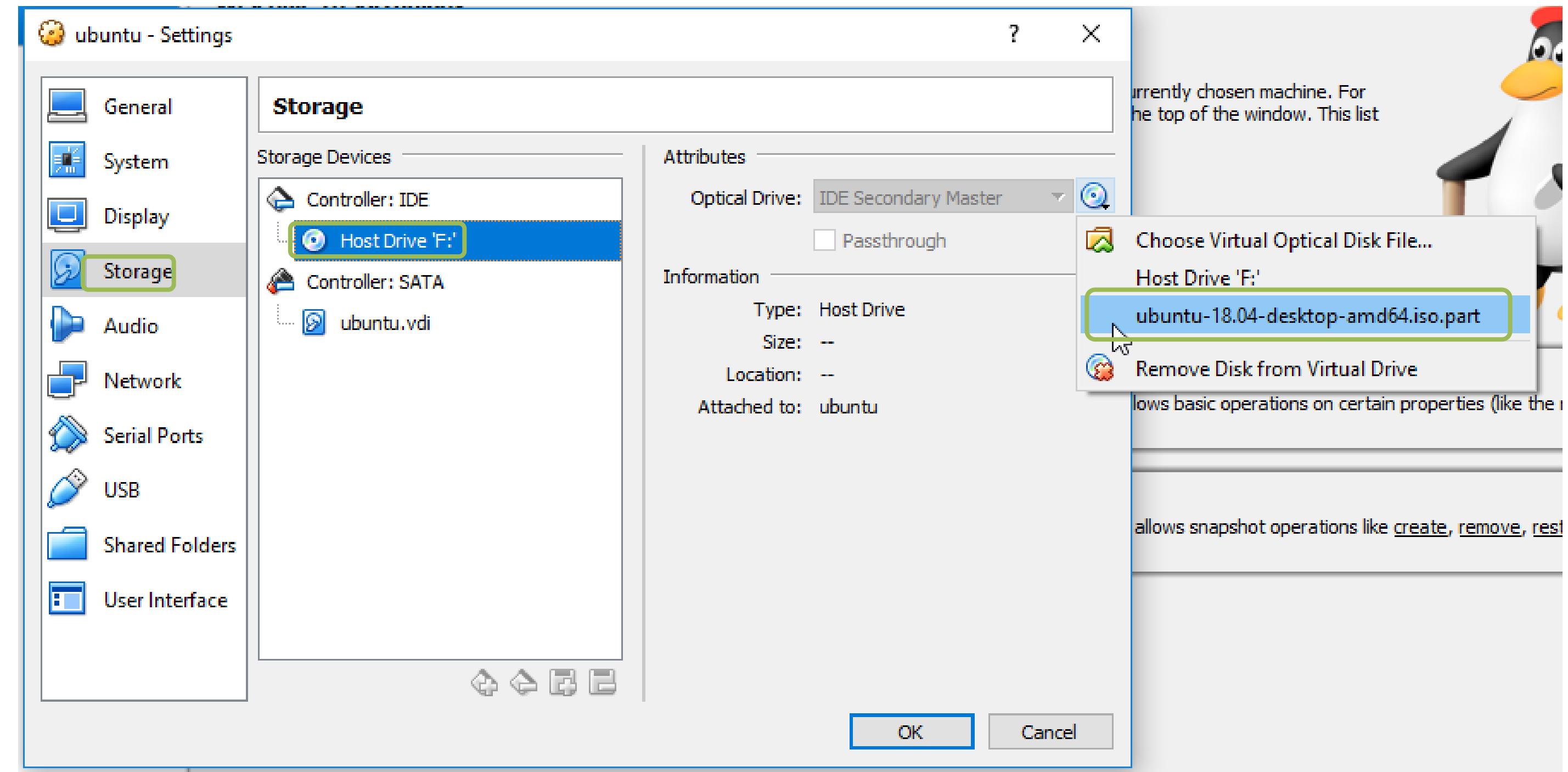
Click on **Settings** on the top left corner. In general settings click on **Advanced** and for both the columns select **Bidirectional**.



Installation And Initialization

Step 17:

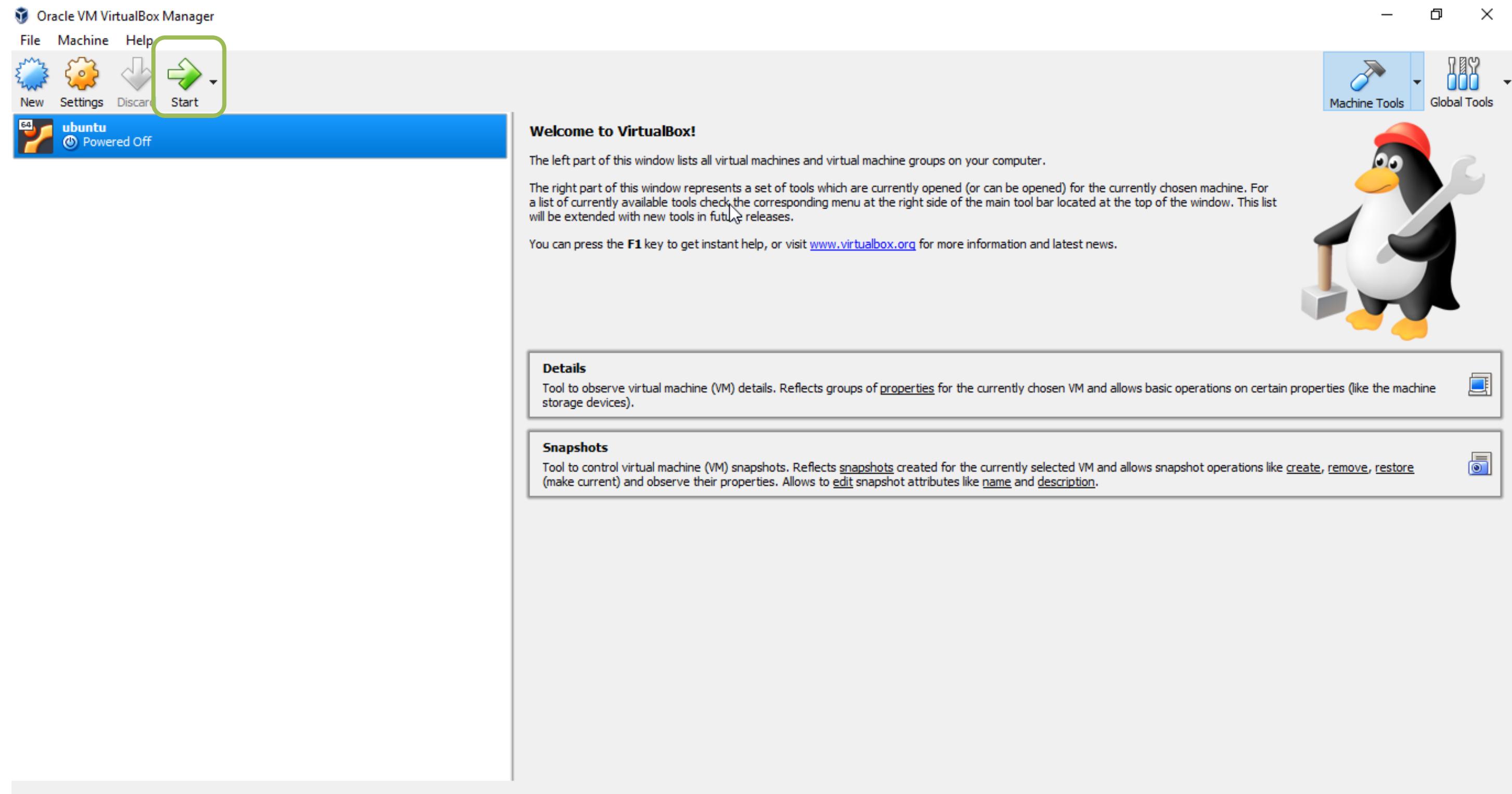
Click on Storage. On selecting **Host Drive 'F'** on the right side you need to click on CD icon and select the **Ubuntu iso** file. If you don't find in the list, you can choose from **Virtual Optical Disk File**.



Installation And Initialization

Step 18:

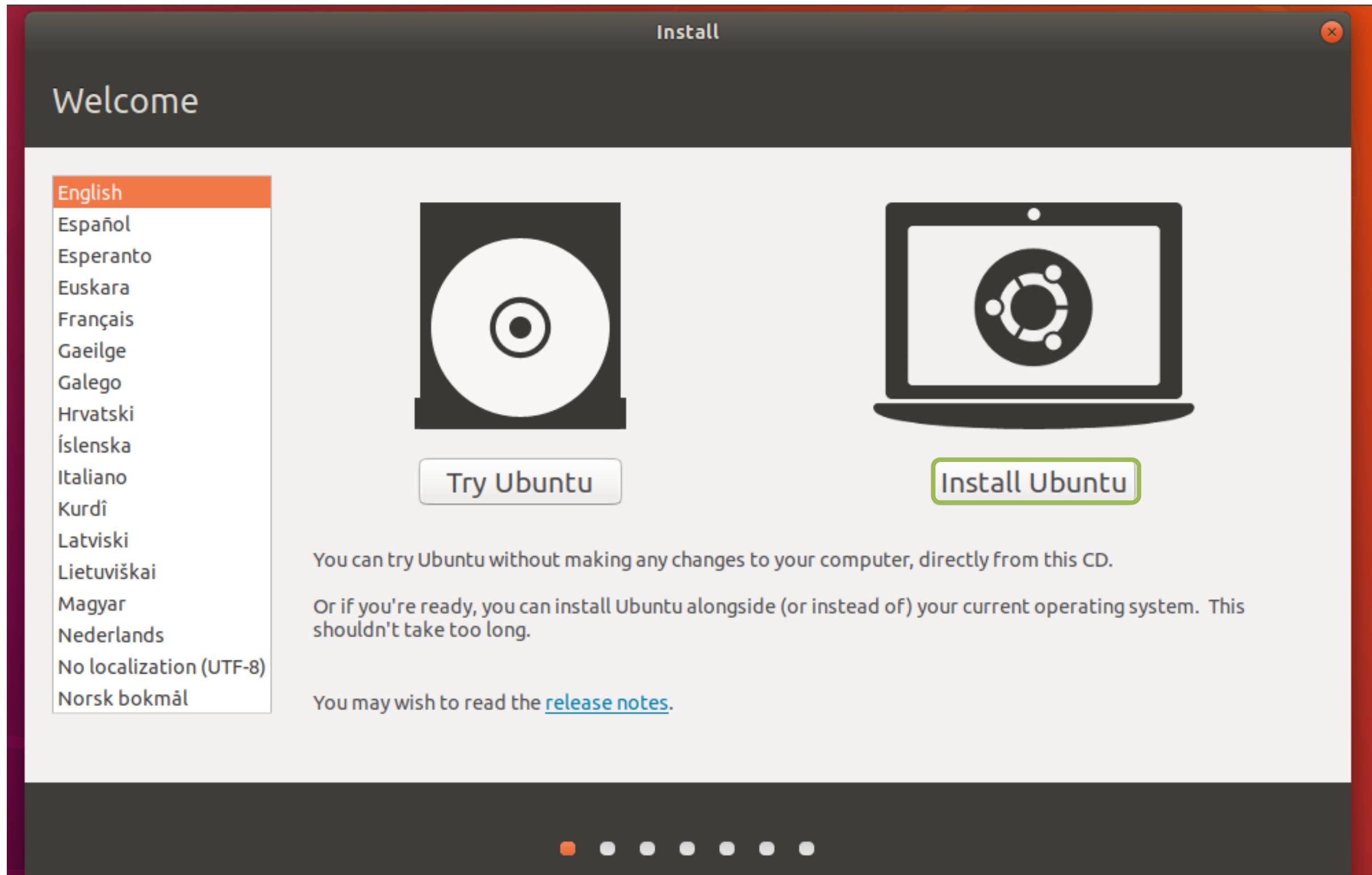
Ubuntu disk image will be created in the left column. Click on **Start**.



Installation And Initialization

Step 19:

Click on **Install Ubuntu**.

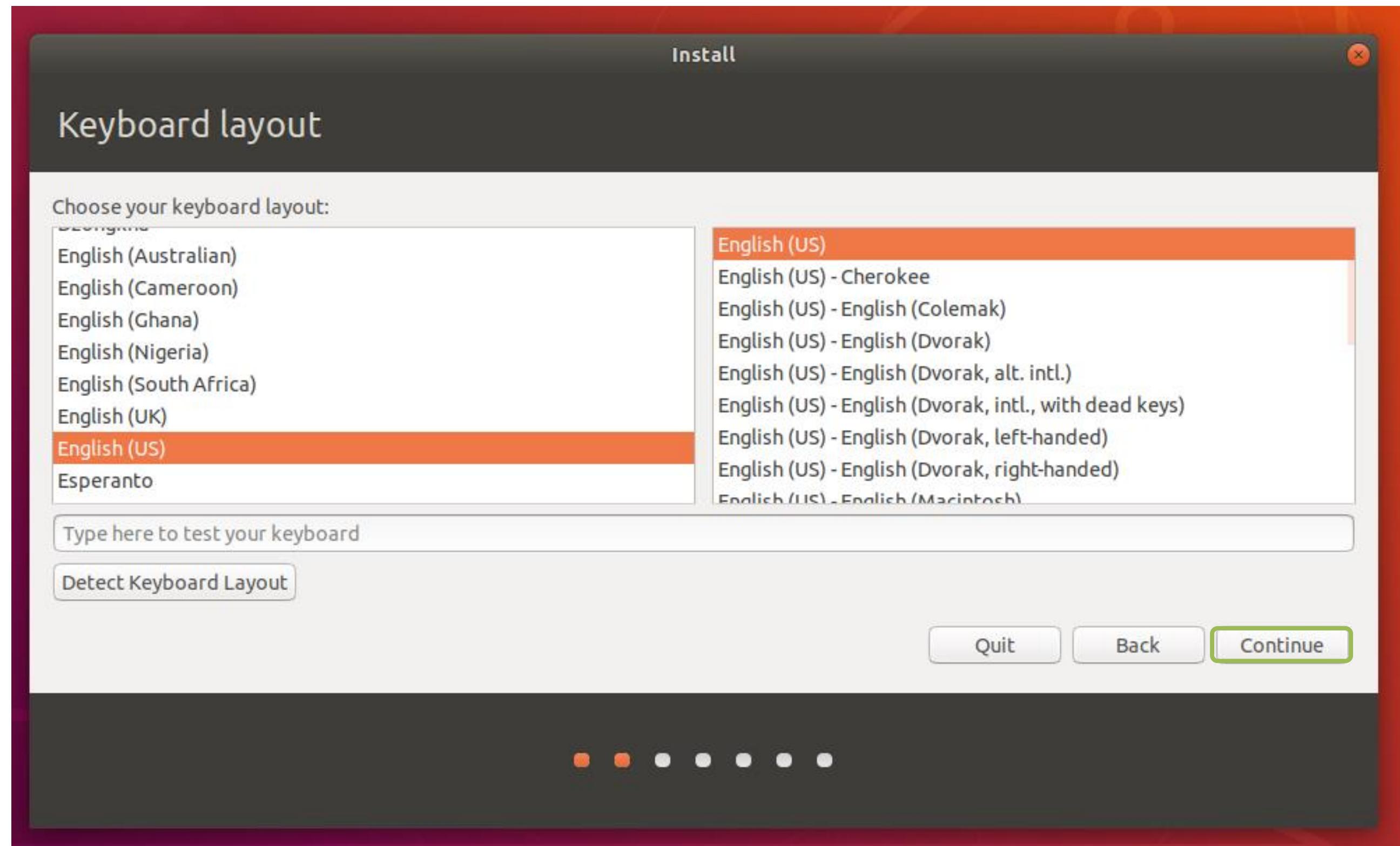


Installation And Initialization

Step 20:

Select any language for Keyboard layout.

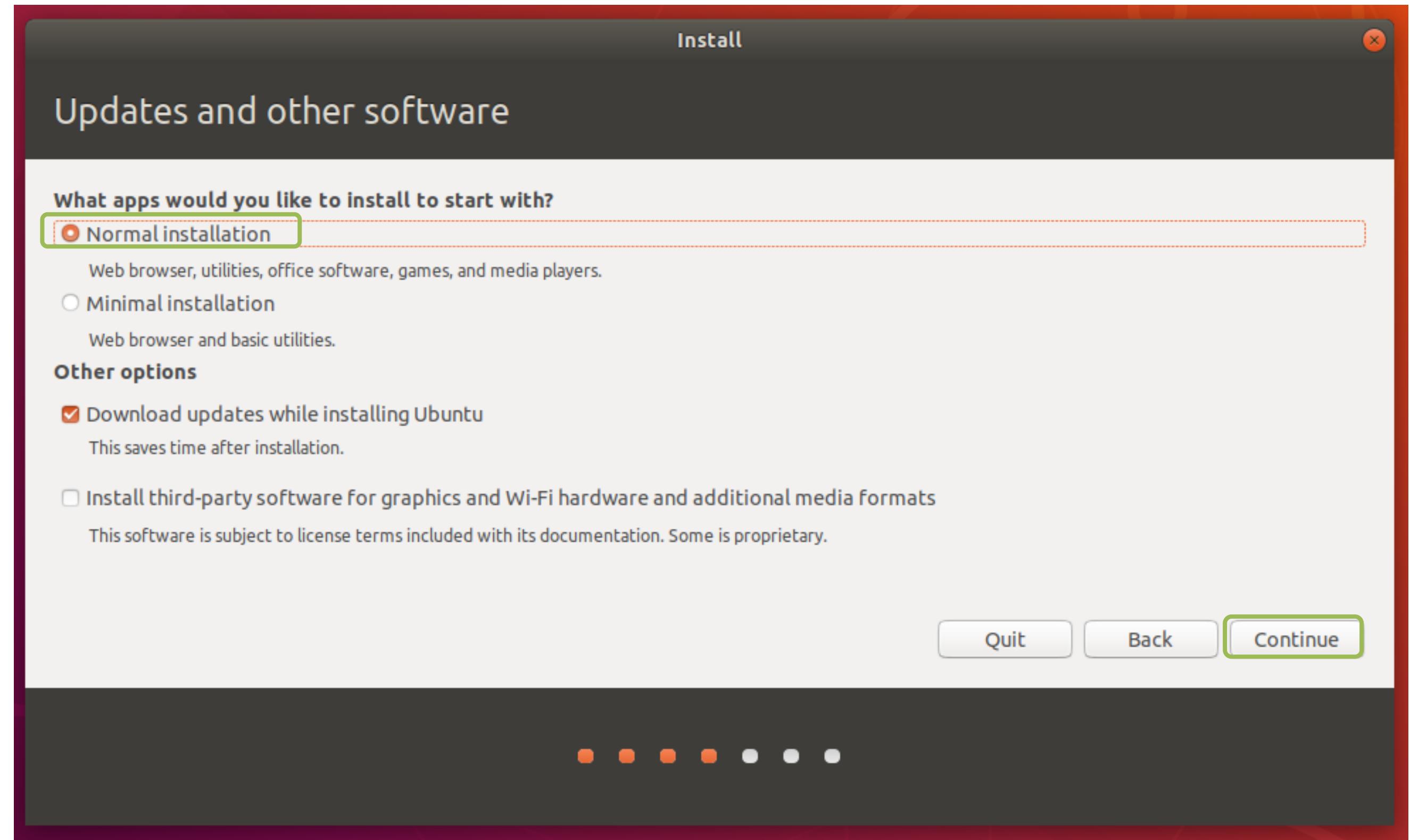
Click on **Continue**.



Installation And Initialization

Step 21:

Select for **Normal Installation**.
Click on **Continue**.

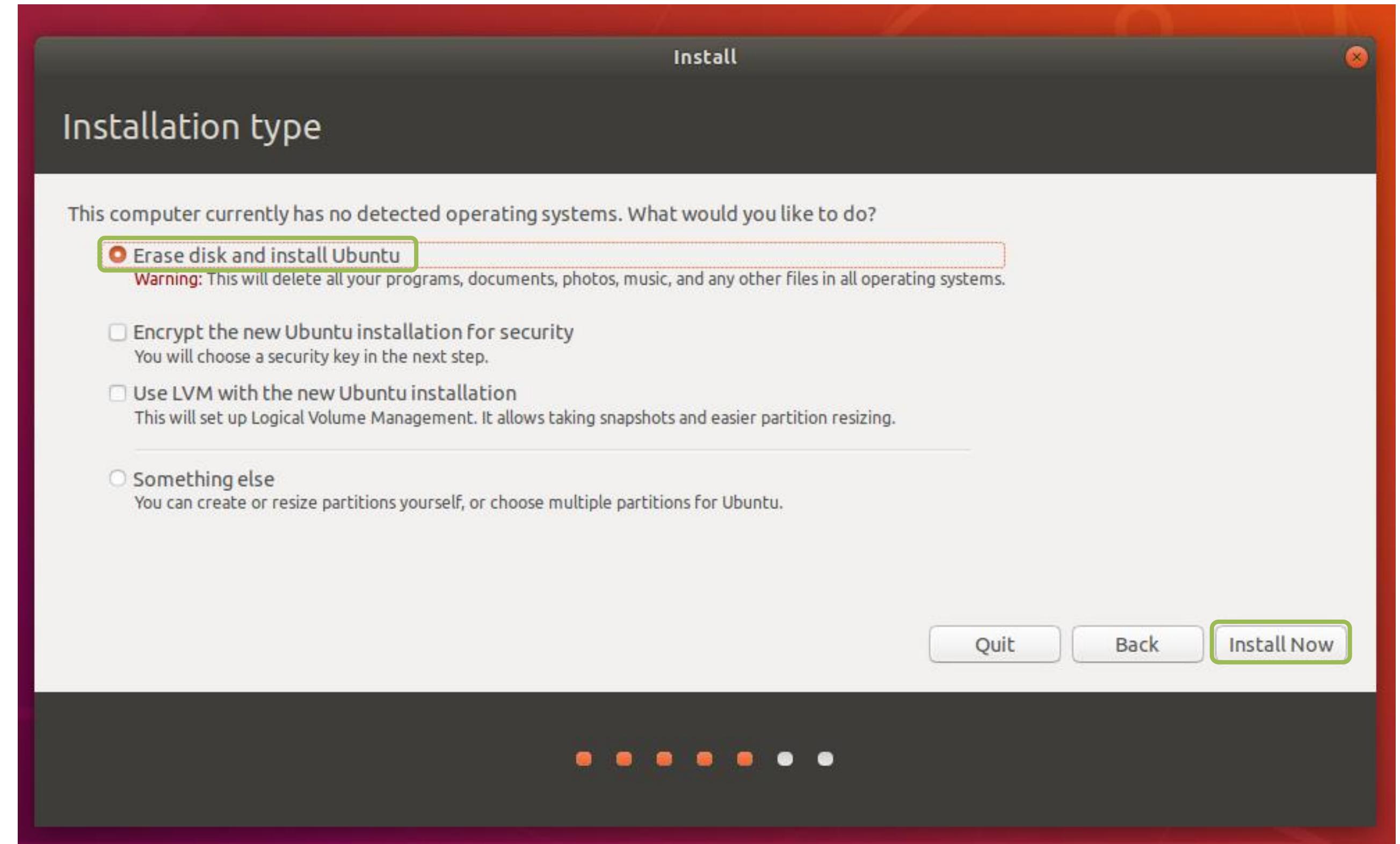


Installation And Initialization

Step 22:

As there will be no any data on disk, you need to select the option **Erase disk and Install Ubuntu**.

Click on **Install Now**.

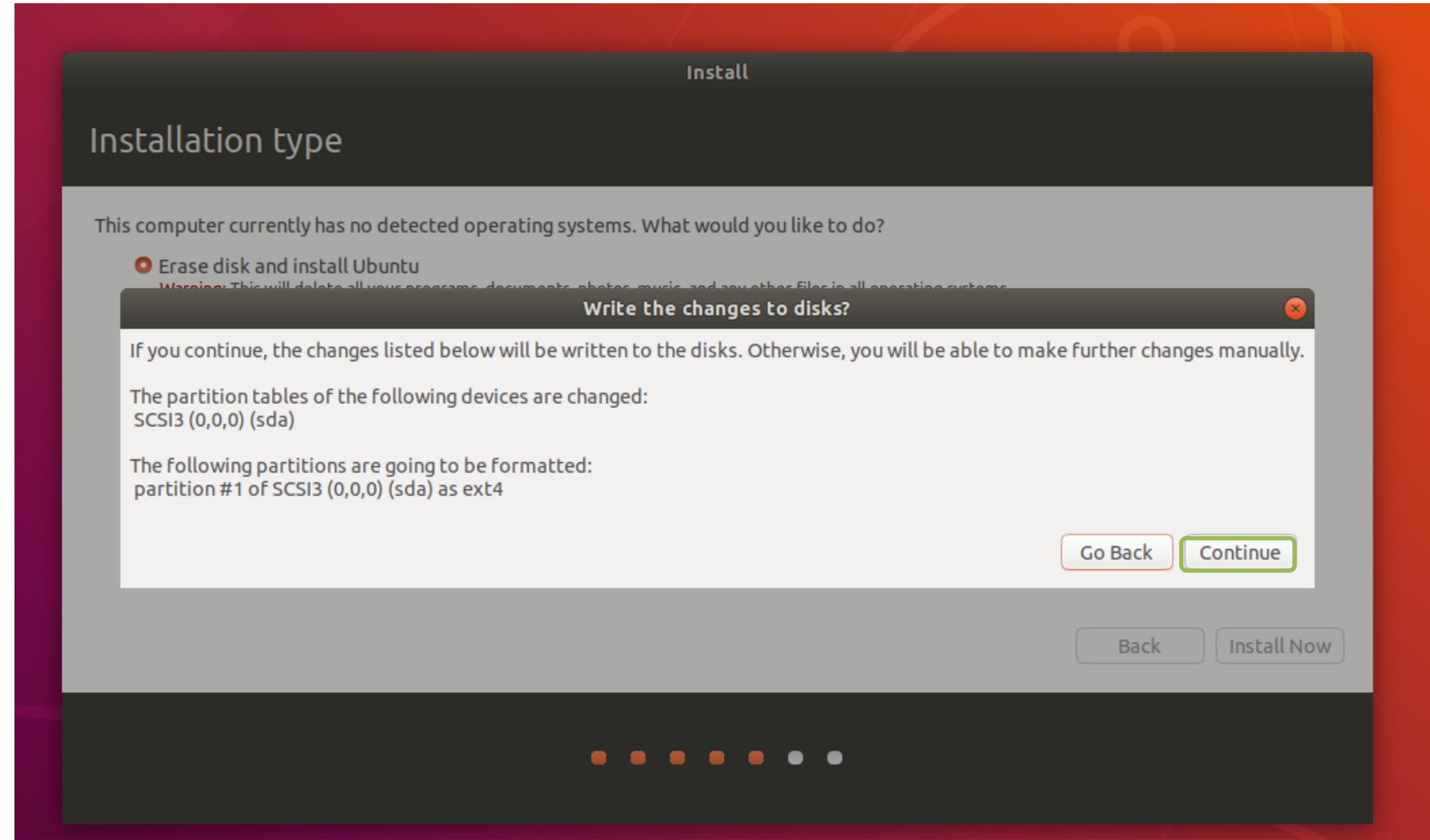


Installation And Initialization

Step 23:

Disk will be portioned here.

Click on **Continue**.

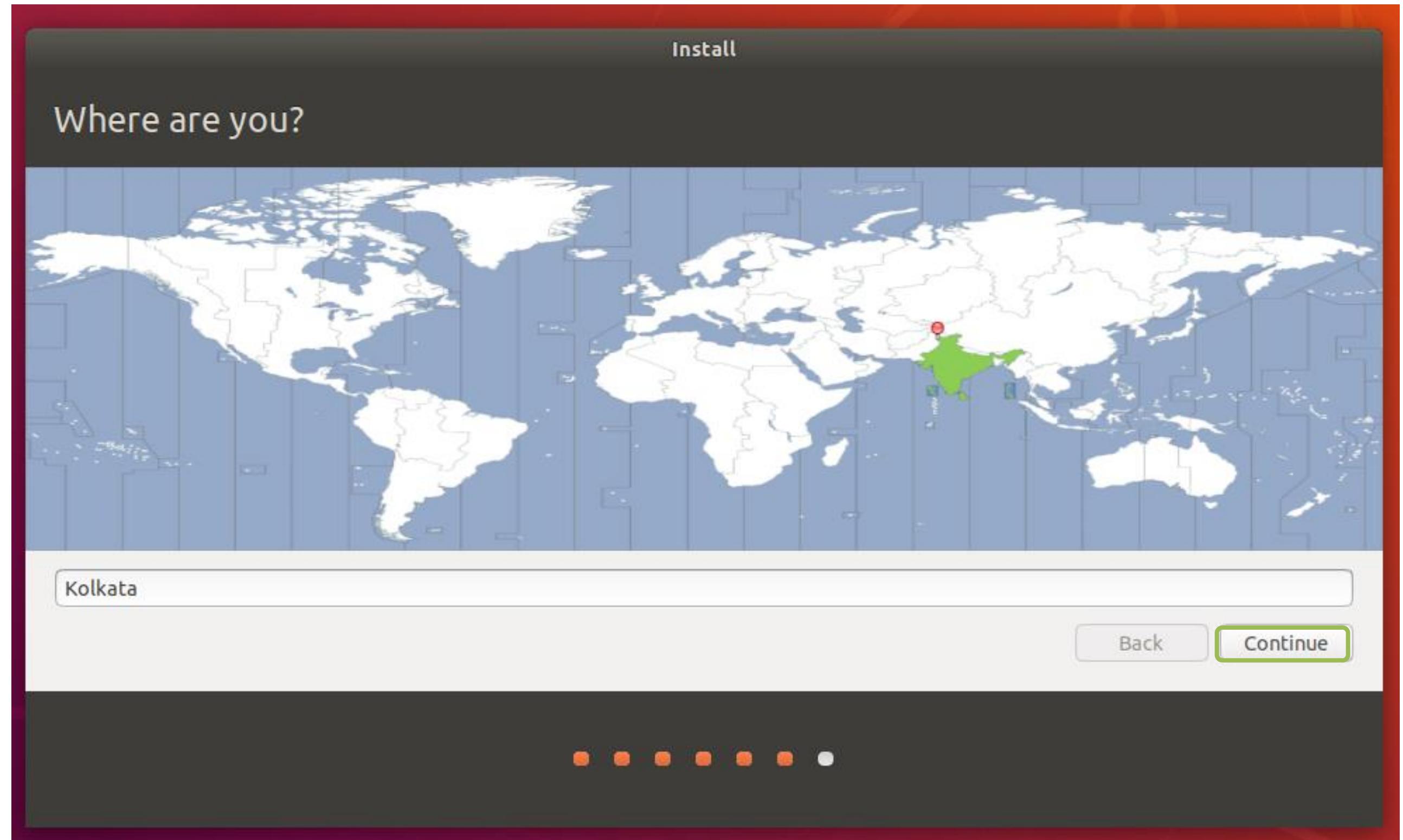


Installation And Initialization

Step 24:

By default your location will be selected.

Click on **Continue**.

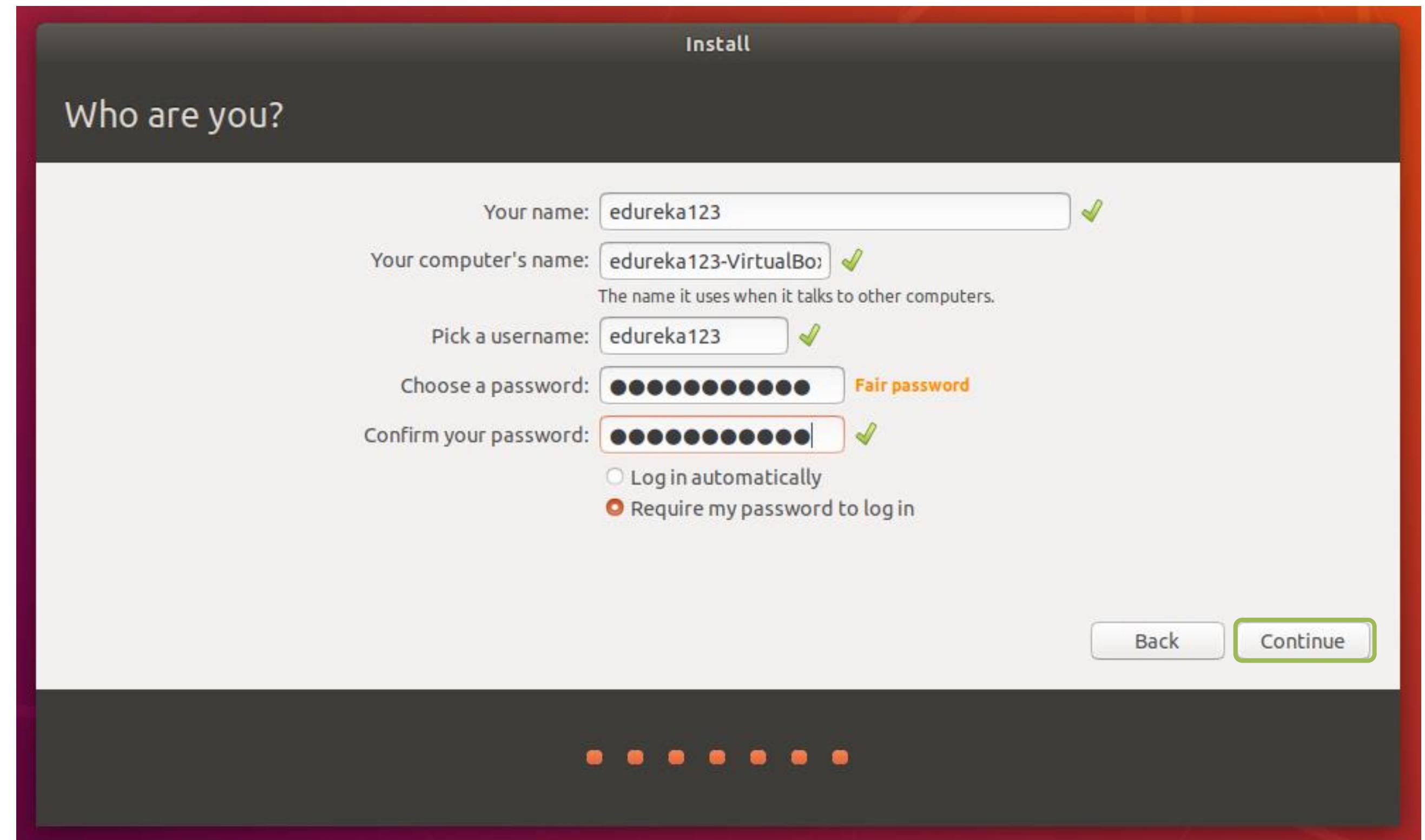


Installation And Initialization

Step 25:

Here fill the details and set the password.

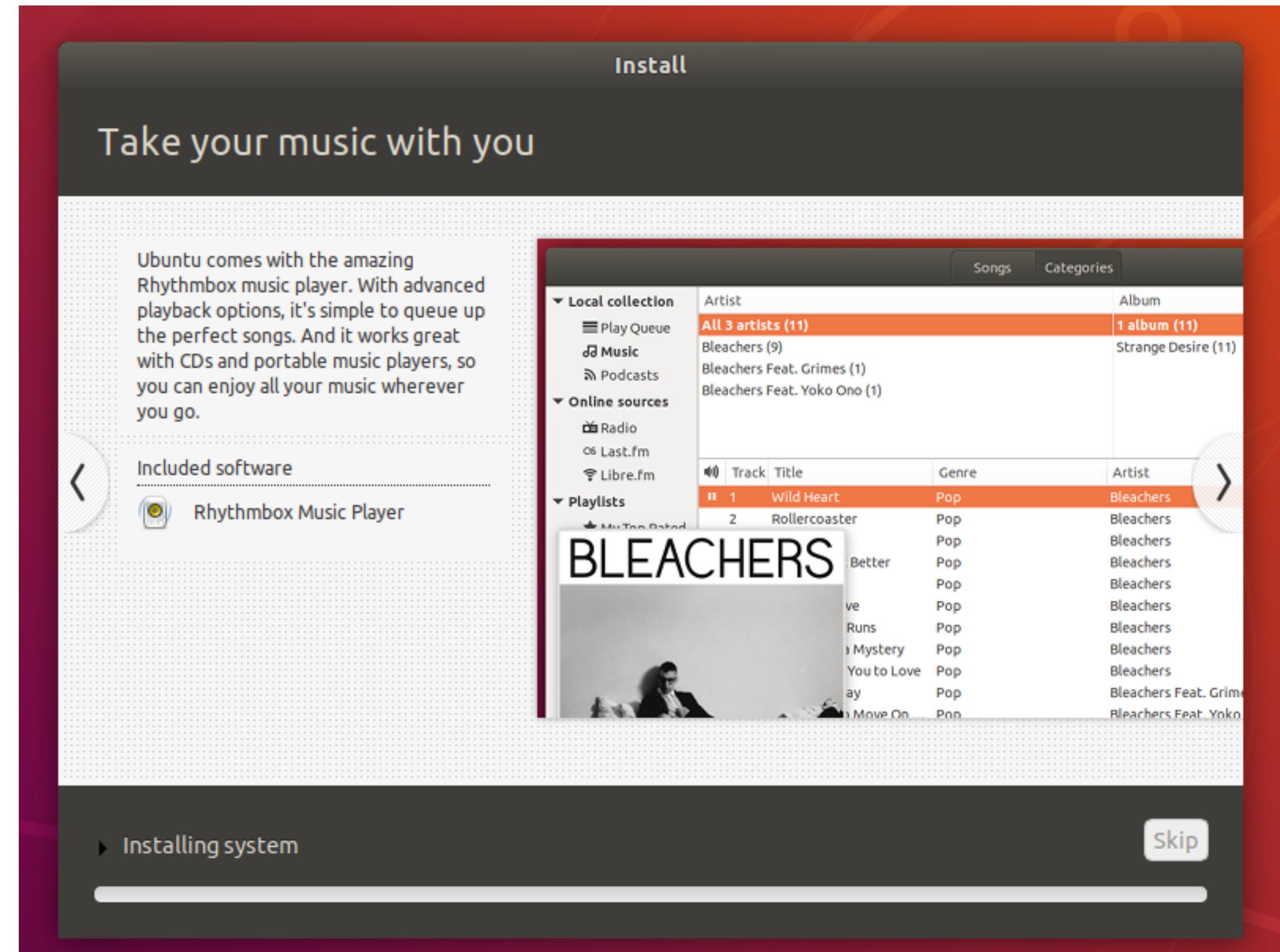
Click on **Continue**.



Installation And Initialization

Step 26:

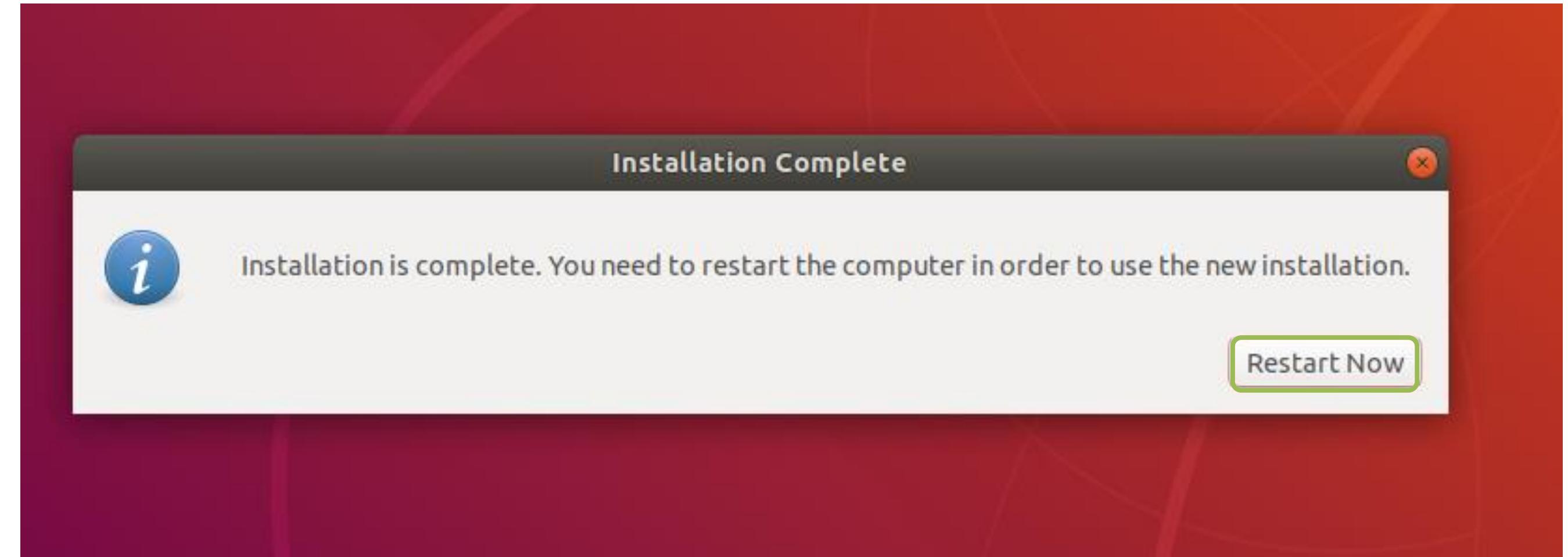
Wait for a while.
Ubuntu will be installed
on your VirtualBox.



Installation And Initialization

Step 27:

Once the process of installation is completed, you need to **Restart**.



Installation And Initialization

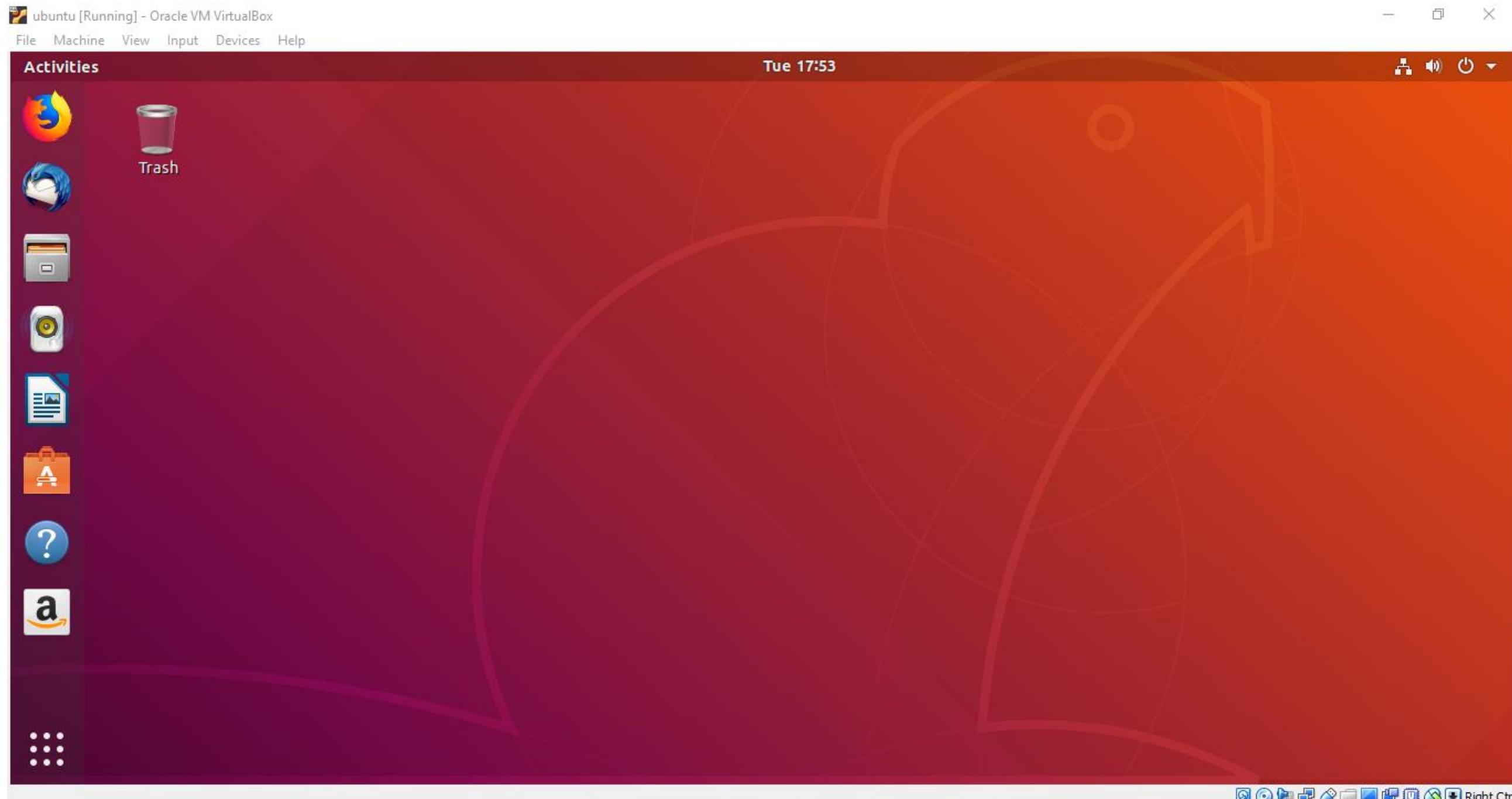
Step 28:

After restarting, enter your **Password**.
Click on **Sign In**.



Installation And Initialization

Here starts the **Ubuntu**, which is one of the Distros of Linux

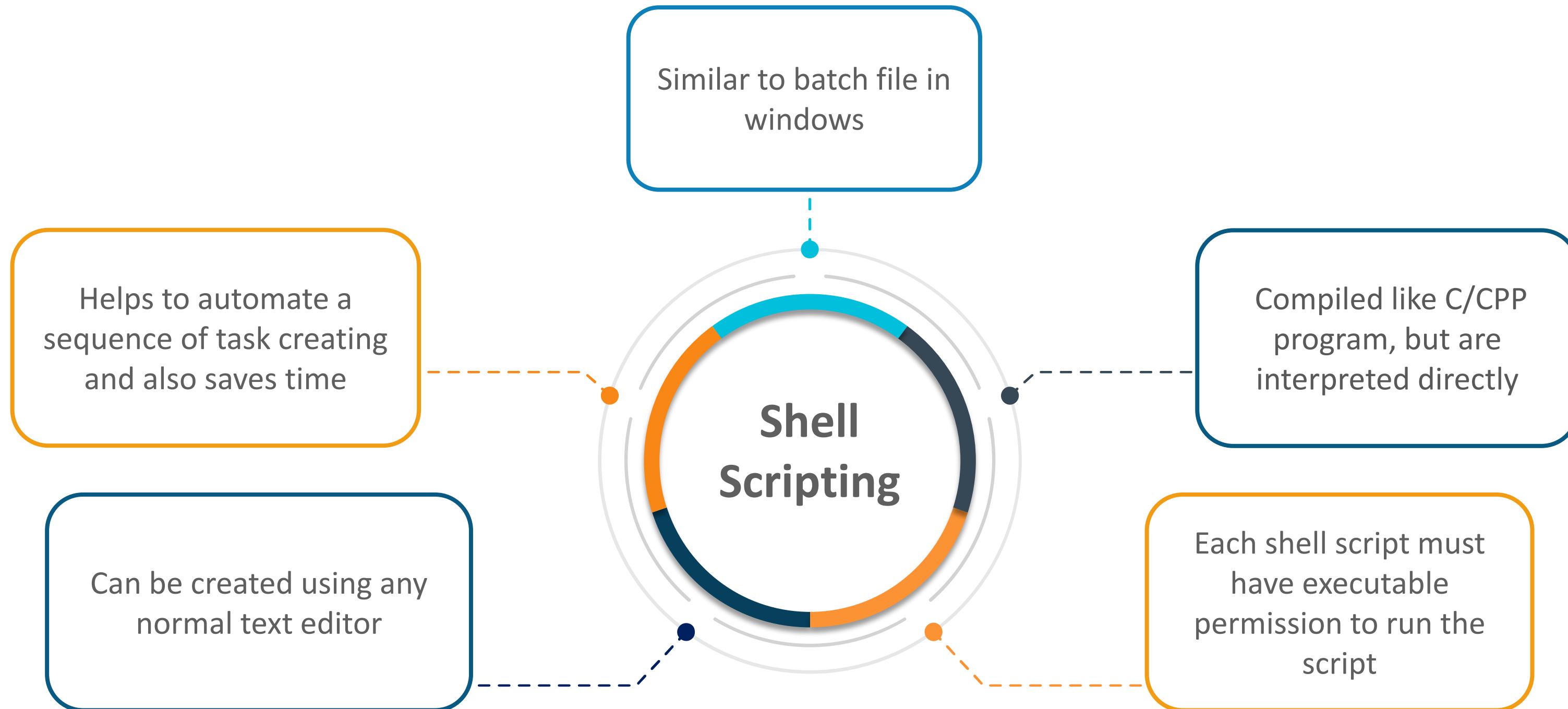




Shell Scripting

Shell Scripting

Shell Scripting is a series of shell commands arranged in a text file to be executed one after the other



Creating A Shell Script

To get into the root for execution

`sudo -i`

Create a text file and add the extension of the shell

`# vi script.sh`

Enter some commands in the scripts to be executed

`echo "My first Script"`
`echo $(date)`

Change the permission of the script, give it executable permission and may also give read and execute permission for other users

`# chmod 755 script.sh`

Run the script

`# ./script.sh`
`# bash script.sh`

Creating A Shell Script

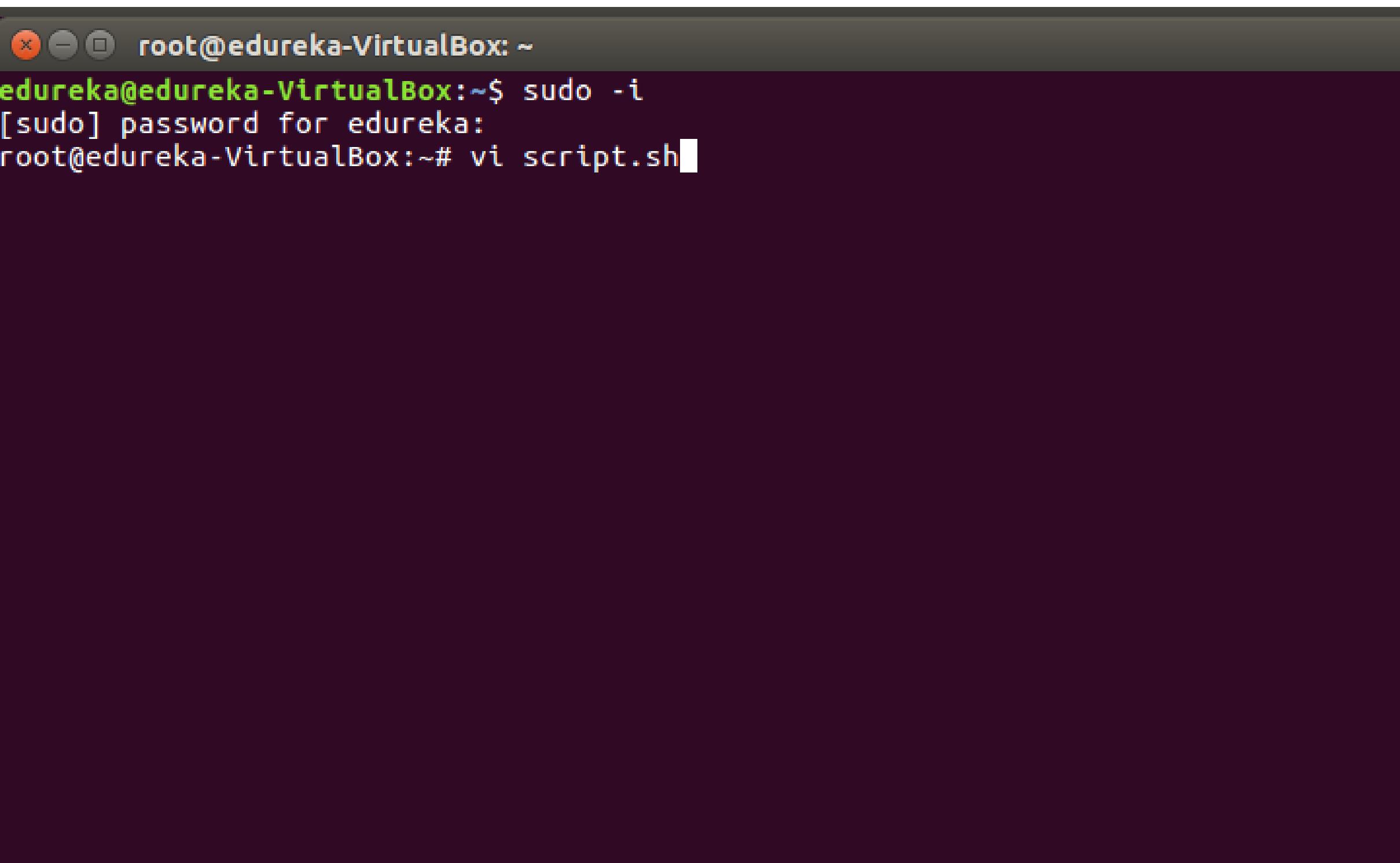
1

```
sudo -i  
# vi script.sh
```

```
echo "My first Script"  
echo $(date)
```

```
# chmod 755 script.sh
```

```
# ./script.sh  
# bash script.sh
```



The screenshot shows a terminal window with a dark background and light-colored text. The window title is 'root@edureka-VirtualBox: ~'. The terminal shows the following sequence of commands:

```
root@edureka-VirtualBox: ~$ sudo -i  
[sudo] password for edureka:  
root@edureka-VirtualBox:~# vi script.sh
```

The 'script.sh' file is created and contains the following content:

```
echo "My first Script"  
echo $(date)
```

After saving the file, the user runs the command:

```
# chmod 755 script.sh
```

Finally, the script is executed:

```
# ./script.sh  
# bash script.sh
```

Creating A Shell Script

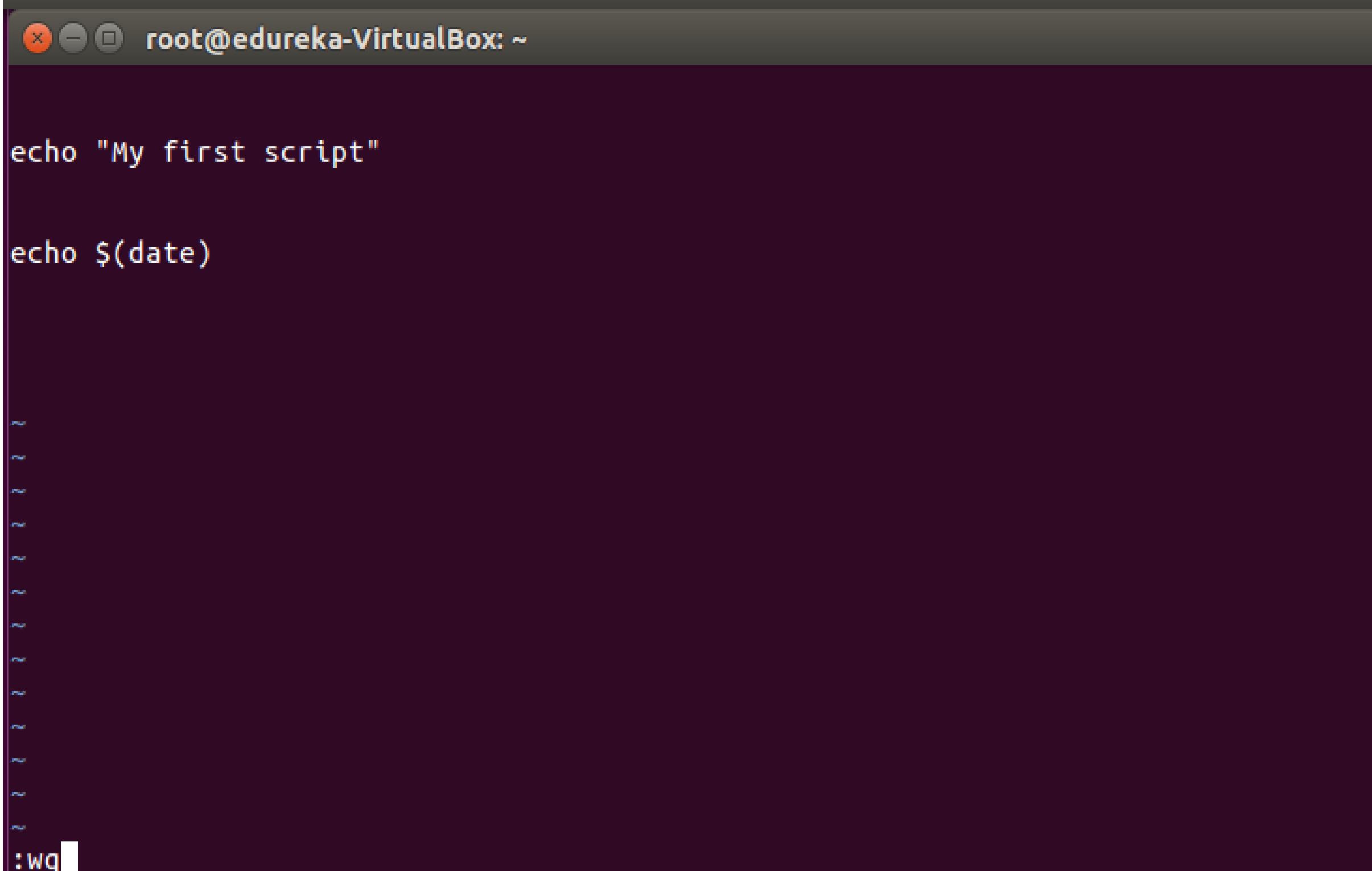
```
sudo -i  
# vi script.sh
```

2

```
echo "My first Script"  
echo $(date)
```

```
# chmod 755 script.sh
```

```
# ./script.sh  
# bash script.sh
```



A terminal window titled "root@edureka-VirtualBox: ~" is shown. The window contains the following text:

```
echo "My first script"  
echo $(date)
```

The terminal window has a dark background and light-colored text. The title bar is dark with light text. The bottom right corner of the terminal window shows the command ":wq" indicating the script is saved and quit.

Creating A Shell Script

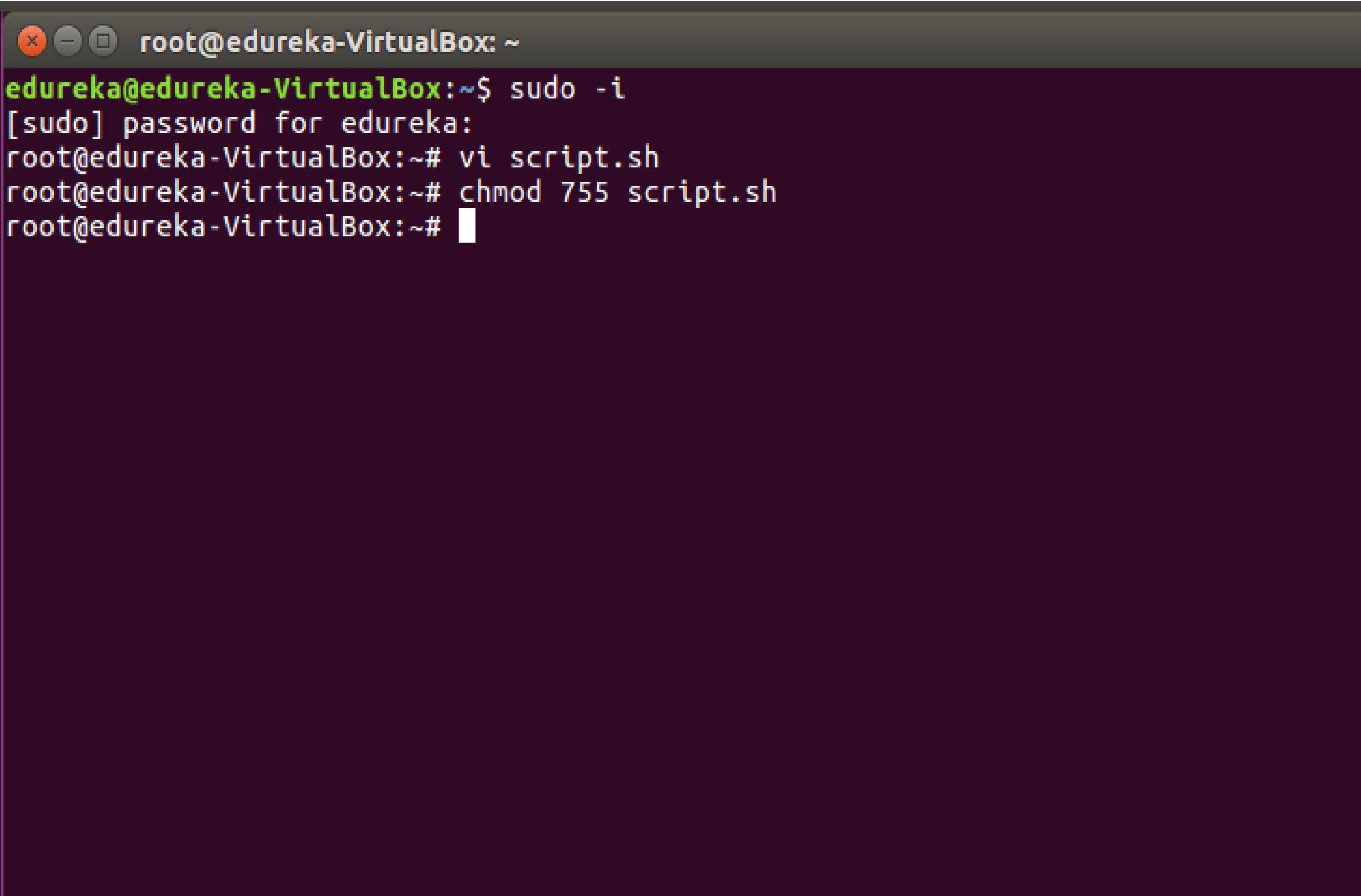
```
sudo -i  
# vi script.sh
```

```
echo "My first Script"  
echo $(date)
```

3

```
# chmod 755 script.sh
```

```
# ./script.sh  
# bash script.sh
```



A terminal window titled 'root@edureka-VirtualBox: ~' showing the creation of a shell script. The commands entered are:

```
root@edureka-VirtualBox: ~$ sudo -i  
[sudo] password for edureka:  
root@edureka-VirtualBox:~# vi script.sh  
root@edureka-VirtualBox:~# chmod 755 script.sh  
root@edureka-VirtualBox:~#
```

Creating A Shell Script

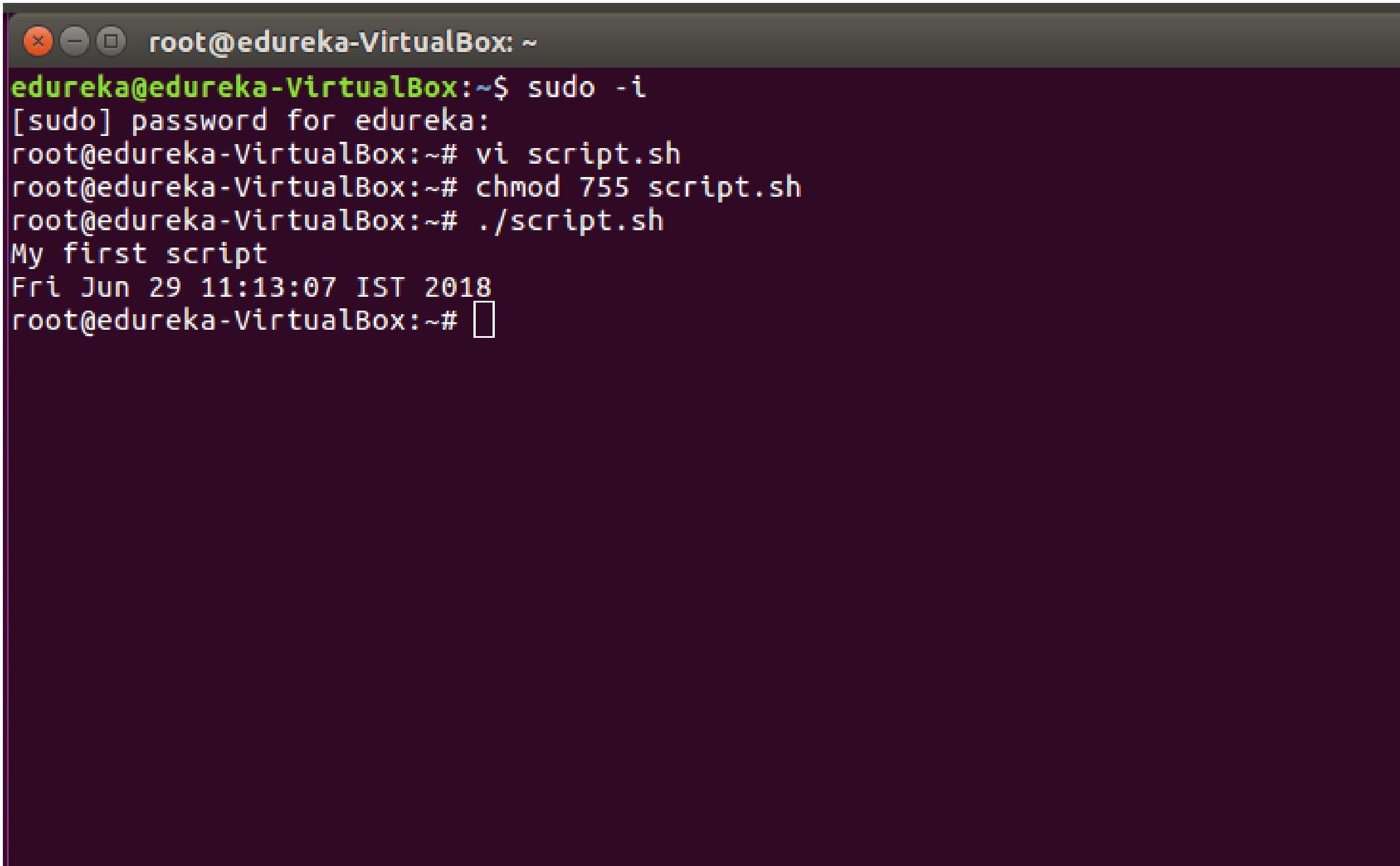
```
sudo -i  
# vi script.sh
```

```
echo "My first Script"  
echo $(date)
```

```
# chmod 755 script.sh
```

4

```
# ./script.sh  
# bash script.sh
```

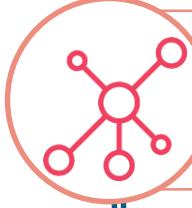


The terminal window shows the following session:

```
root@edureka-VirtualBox: ~  
edureka@edureka-VirtualBox:~$ sudo -i  
[sudo] password for edureka:  
root@edureka-VirtualBox:~# vi script.sh  
root@edureka-VirtualBox:~# chmod 755 script.sh  
root@edureka-VirtualBox:~# ./script.sh  
My first script  
Fri Jun 29 11:13:07 IST 2018  
root@edureka-VirtualBox:~#
```

Practical Uses Of Shell Scripting



-  Data backup at regular interval of time in the background
-  To find out the number of users and their details
-  Find details about various processes and sorting them based on CPU usage, runtime, memory usage, users, etc.
-  To append each file with a signature, date, etc.
-  For scheduling tasks to be done at network servers
-  Creating new users by providing permissions just by entering username

Summary

- In this module, you should have learned:
 - Linux and its development
 - Components of a Linux distro and which one to choose
 - Some of the Linux OS concepts
 - Various kinds of Software Licensing



Questions



Thank You



For more information please visit our website
www.edureka.co