

The Unix operating system is a set of programs that act as a link between the computer and the user.

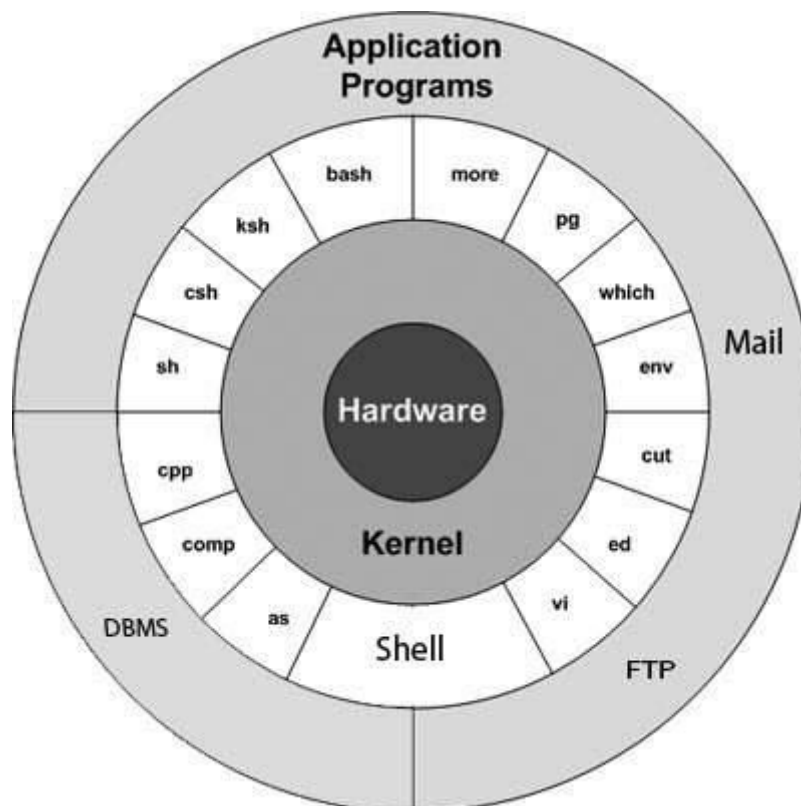
The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **operating system** or the **kernel**.

Users communicate with the kernel through a program known as the **shell**. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

- Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.
- There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.
- Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.
- A user can also run multiple programs at the same time; hence Unix is a multitasking environment.

## Unix Architecture

Here is a basic block diagram of a Unix system –



The main concept that unites all the versions of Unix is the following four basics –

- **Kernel** – The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

- **Shell** – The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.
- **Commands and Utilities** – There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3<sup>rd</sup> party software. All the commands come along with various options.
- **Files and Directories** – All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.

## System Bootup

If you have a computer which has the Unix operating system installed in it, then you simply need to turn on the system to make it live.

As soon as you turn on the system, it starts booting up and finally it prompts you to log into the system, which is an activity to log into the system and use it for your day-to-day activities.

## Login Unix

When you first connect to a Unix system, you usually see a prompt such as the following –

```
login:
```

## To log in

- Have your userid (user identification) and password ready. Contact your system administrator if you don't have these yet.
- Type your userid at the login prompt, then press **ENTER**. Your userid is **case-sensitive**, so be sure you type it exactly as your system administrator has instructed.
- Type your password at the password prompt, then press **ENTER**. Your password is also case-sensitive.
- If you provide the correct userid and password, then you will be allowed to enter into the system. Read the information and messages that comes up on the screen, which is as follows.

```
login : amrood
amrood's password:
Last login: Sun Jun 14 09:32:32 2009 from 62.61.164.73
$
```

You will be provided with a command prompt (sometime called the **\$** prompt ) where you type all your commands. For example, to check calendar, you need to type the **cal** command as follows –

```
$ cal
      June 2009
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

```
$
```

## Change Password

All Unix systems require passwords to help ensure that your files and data remain your own and that the system itself is secure from hackers and crackers. Following are the steps to change your password –

**Step 1** – To start, type password at the command prompt as shown below.

**Step 2** – Enter your old password, the one you're currently using.

**Step 3** – Type in your new password. Always keep your password complex enough so that nobody can guess it. But make sure, you remember it.

**Step 4** – You must verify the password by typing it again.

```
$ passwd
Changing password for amrood
(current) Unix password:*****
New UNIX password:*****
Retype new UNIX password:*****
passwd: all authentication tokens updated successfully
```

```
$
```

**Note** – We have added asterisk (\*) here just to show the location where you need to enter the current and new passwords otherwise at your system. It does not show you any character when you type.

## Listing Directories and Files

All data in Unix is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the filesystem.

You can use the **ls** command to list out all the files or directories available in a directory. Following is the example of using **ls** command with **-l** option.

```
$ ls -l
total 19621
drwxrwxr-x  2 amrood amrood      4096 Dec 25 09:59 uml
-rw-rw-r--  1 amrood amrood     5341 Dec 25 08:38 uml.jpg
drwxr-xr-x  2 amrood amrood      4096 Feb 15  2006 univ
drwxr-xr-x  2 root   root        4096 Dec  9  2007 urlspedia
-rw-r--r--  1 root   root     276480 Dec  9  2007 urlspedia.tar
drwxr-xr-x  8 root   root        4096 Nov 25  2007 usr
-rwxr-xr-x  1 root   root        3192 Nov 25  2007 webthumb.php
-rw-rw-r--  1 amrood amrood    20480 Nov 25  2007 webthumb.tar
```

```
-rw-rw-r-- 1 amrood amrood      5654 Aug  9  2007 yourfile.mid  
-rw-rw-r-- 1 amrood amrood    166255 Aug  9  2007 yourfile.swf
```

\$

Here entries starting with **d.....** represent directories. For example, `uml`, `univ` and `urlspedia` are directories and rest of the entries are files.

## Who Are You?

While you're logged into the system, you might be willing to know : **Who am I?**

The easiest way to find out "who you are" is to enter the **whoami** command –

```
$ whoami  
amrood
```

\$

Try it on your system. This command lists the account name associated with the current login. You can try **who am i** command as well to get information about yourself.

## Who is Logged in?

Sometime you might be interested to know who is logged in to the computer at the same time.

There are three commands available to get you this information, based on how much you wish to know about the other users: **users**, **who**, and **w**.

```
$ users  
amrood bablu qadir
```

```
$ who  
amrood ttyp0 Oct 8 14:10 (limbo)  
bablu ttyp2 Oct 4 09:08 (calliope)  
qadir ttyp4 Oct 8 12:09 (dent)
```

\$

Try the **w** command on your system to check the output. This lists down information associated with the users logged in the system.

## Logging Out

When you finish your session, you need to log out of the system. This is to ensure that nobody else accesses your files.

### To log out

- Just type the **logout** command at the command prompt, and the system will clean up everything and break the connection.

## System Shutdown

The most consistent way to shut down a Unix system properly via the command line is to use one of the following commands –

Sr.No	Command & Description
1	<b>halt</b> Brings the system down immediately
2	<b>init 0</b> Powers off the system using predefined scripts to synchronize and clean up the system prior to shutting down
3	<b>init 6</b> Reboots the system by shutting it down completely and then restarting it
4	<b>poweroff</b> Shuts down the system by powering off
5	<b>reboot</b> Reboots the system
6	<b>shutdown</b> Shuts down the system

You typically need to be the super user or root (the most privileged account on a Unix system) to shut down the system. However, on some standalone or personally-owned Unix boxes, an administrative user and sometimes regular users can do so.

Year	Event
1957	Bell Labs found they needed an operating system for their computer center which at the time was running various batch jobs. The BESYS operating system was created at Bell Labs to deal with these needs.
1965	Bell Labs was adopting third generation computer equipment and decided to join forces with General Electric and MIT to create Multics (Multiplexed Information and Computing Service).
1969	By April 1969, <a href="#">AT&amp;T</a> made a decision to withdraw Multics and go with GECOS. When Multics was withdrawn Ken Thompson and Dennis Ritchie needed to rewrite an operating system in order to play space travel on another smaller machine (a DEC

	PDP-7 [Programmed Data Processor 4K memory for user programs). The result was a system which a punning colleague called UNICS (UNiplexed Information and Computing Service)--an 'emasculated Multics'.
<b>1969</b>	Summer 1969 UNIX was developed.
<b>1971</b>	First edition of <a href="#">UNIX</a> released 11/03/1971. The first edition of the "UNIX PROGRAMMER'S MANUAL [by] K. Thompson [and] D. M. Ritchie" is also dated "November 3, 1971". It includes over 60 commands like: b (compile B program); boot (reboot system); <a href="#">cat</a> (concatenate files); <a href="#">chdir</a> (change working directory); <a href="#">chmod</a> (change access mode); <a href="#">chown</a> (change owner); <a href="#">cp</a> (copy file); <a href="#">ls</a> (list directory contents); <a href="#">mv</a> (move or rename file); roff (run off text); <a href="#">wc</a> (get word count); <a href="#">who</a> (who is one the system). The main thing missing was pipes.
<b>1972</b>	Second edition of <a href="#">UNIX</a> released 12/06/1972
<b>1972</b>	Ritchie rewrote B and called the new language C.
<b>1973</b>	<a href="#">UNIX</a> had been installed on 16 sites (all within AT&T/Western Electric); it was publically unveiled at a conference in October.
<b>1973</b>	Third edition of <a href="#">UNIX</a> released 02/xx/1973
<b>1973</b>	Forth edition of <a href="#">UNIX</a> released 11/xx/1973
<b>1974</b>	Fifth edition of <a href="#">UNIX</a> released 06/xx/1974

These commands allow you to create directories and handle files.

Given below is the list of commands in Files and Directories.

Sr.No.	Command & Description
1	<b>cat</b> Displays File Contents
2	<b>cd</b> Changes Directory to dirname
3	<b>chgrp</b> Changes file group
4	<b>chmod</b> Changes permissions
5	<b>cp</b> Copies source file into destination

6	<b>file</b> Determines file type
7	<b>find</b> Finds files
8	<b>grep</b> Searches files for regular expressions
9	<b>head</b> Displays first few lines of a file
10	<b>ln</b> Creates softlink on oldname
11	<b>ls</b> Displays information about file type
12	<b>mkdir</b> Creates a new directory dirname
13	<b>more</b> Displays data in paginated form
14	<b>mv</b> Moves (Renames) an oldname to newname
15	<b>pwd</b> Prints current working directory
16	<b>rm</b> Removes (Deletes) filename
17	<b>rmdir</b> Deletes an existing directory provided it is empty

18	<b>tail</b> Prints last few lines in a file
19	<b>touch</b> Updates access and modification time of a file

## Manipulating data

The contents of files can be compared and altered with the following commands.

Given below is the list of commands in Manipulating data.

Sr.No.	Command & Description
1	<b>awk</b> Pattern scanning and processing language
2	<b>cmp</b> Compares the contents of two files
3	<b>comm</b> Compares sorted data
4	<b>cut</b> Cuts out selected fields of each line of a file
5	<b>diff</b> Differential file comparator
6	<b>expand</b> Expands tabs to spaces
7	<b>join</b> Joins files on some common field
8	<b>perl</b>



	Data manipulation language
9	<b>sed</b> Stream text editor
10	<b>sort</b> Sorts file data
11	<b>split</b> Splits file into smaller files
12	<b>tr</b> Translates characters
13	<b>uniq</b> Reports repeated lines in a file
14	<b>wc</b> Counts words, lines, and characters
15	<b>vi</b> Opens vi text editor
16	<b>vim</b> Opens vim text editor
17	<b>fmt</b> Simple text formatter
18	<b>spell</b> Checks text for spelling error
19	<b>ispell</b> Checks text for spelling error

20	<b>emacs</b> GNU project Emacs
21	<b>ex, edit</b> Line editor
22	<b>emacs</b> GNU project Emacs

## Compressed Files

Files may be compressed to save space. Compressed files can be created and examined.

Sr.No.	Command & Description
1	<b>Compress</b> Compresses files
2	<b>Gunzip</b> Helps uncompress gzipped files
3	<b>Gzip</b> GNU alternative compression method
4	<b>Uncompress</b> Helps uncompress files
5	<b>Unzip</b> List, test and extract compressed files in a ZIP archive
6	<b>Zcat</b> Cat a compressed file
7	<b>Zcmp</b>

	Compares compressed files
8	<b>Zdiff</b> Compares compressed files
9	<b>Zmore</b> File perusal filter for crt viewing of compressed text

## Getting Information

Various Unix manuals and documentation are available on-line. The following Shell commands give information –

Sr.No.	Command & Description
1	<b>Apropos</b> Locates commands by keyword lookup
2	<b>Info</b> Displays command information pages online
2	<b>Man</b> Displays manual pages online
3	<b>Whatis</b> Searches the whatis database for complete words
4	<b>Yelp</b> GNOME help viewer

## Network Communication

These following commands are used to send and receive files from a local Unix hosts to the remote host around the world.

Sr.No.	Command & Description
--------	-----------------------

1	<b>ftp</b> File transfer program
2	<b>Rcp</b> Remote file copy
3	<b>rlogin</b> Remote login to a Unix host
4	<b>Rsh</b> Remote shell
5	<b>Tftp</b> Trivial file transfer program
6	<b>telnet</b> Makes terminal connection to another host
7	<b>Ssh</b> Secures shell terminal or command connection
8	<b>Scp</b> Secures shell remote file copy
9	<b>Sftp</b> Secures shell file transfer program

Some of these commands may be restricted at your computer for security reasons.

## Messages between Users

The Unix systems support on-screen messages to other users and world-wide electronic mail –

Sr.No.	Command & Description
--------	-----------------------

1	<b>Evolution</b> GUI mail handling tool on Linux
2	<b>Mail</b> Simple send or read mail program
3	<b>Mesg</b> Permits or denies messages
4	<b>Parcel</b> Sends files to another user
5	<b>Pine</b> Vdu-based mail utility
6	<b>Talk</b> Talks to another user
7	<b>Write</b> Writes message to another user

## Programming Utilities

The following programming tools and languages are available based on what you have installed on your Unix.

Given below is the list of tools and languages in Programming Utilities.

Sr.No.	Command & Description
1	<b>Dbx</b> Sun debugger
2	<b>Gdb</b> GNU debugger
3	<b>Make</b>

	Maintains program groups and compile programs
4	<b>Nm</b> Prints program's name list
5	<b>Size</b> Prints program's sizes
6	<b>Strip</b> Removes symbol table and relocation bits
7	<b>Cb</b> C program beautifier
8	<b>Cc</b> ANSI C compiler for Suns SPARC systems
9	<b>Ctrace</b> C program debugger
10	<b>Gcc</b> GNU ANSI C Compiler
11	<b>Indent</b> Indent and format C program source
12	<b>Bc</b> Interactive arithmetic language processor
13	<b>Gcl</b> GNU Common Lisp
14	<b>Perl</b> General purpose language

15	<b>Php</b> Web page embedded language
16	<b>Py</b> Python language interpreter
17	<b>Asp</b> Web page embedded language
18	<b>CC</b> C++ compiler for Suns SPARC systems
19	<b>g++</b> GNU C++ Compiler
20	<b>Javac</b> JAVA compiler
21	<b>Appletvieweir</b> JAVA applet viewer
22	<b>Netbeans</b> Java integrated development environment on Linux
23	<b>Sqlplus</b> Runs the Oracle SQL interpreter
24	<b>Sqldr</b> Runs the Oracle SQL data loader
25	<b>Mysql</b> Runs the mysql SQL interpreter

## Misc Commands

These commands list or alter information about the system –

Given below is the list of Misc Commands in Unix.

Sr.No.	Command & Description
1	<b>Chfn</b> Changes your finger information
2	<b>Chgrp</b> Changes the group ownership of a file
3	<b>Chown</b> Changes owner
4	<b>Date</b> Prints the date
5	<b>Determin</b> Automatically finds terminal type
6	<b>Du</b> Prints amount of disk usage
7	<b>Echo</b> Echo arguments to the standard options
8	<b>Exit</b> Quits the system
9	<b>Finger</b> Prints information about logged-in users
10	<b>Groupadd</b> Creates a user group
11	<b>Groups</b>



	Show group memberships
12	<b>Homequota</b> Shows quota and file usage
13	<b>Iostat</b> Reports I/O statistics
14	<b>Kill</b> Sends a signal to a process
15	<b>Last</b> Shows last logins of users
16	<b>Logout</b> Logs off Unix
17	<b>Lun</b> Lists user names or login ID
18	<b>Netstat</b> Shows network status
19	<b>Passwd</b> Changes user password
20	<b>Passwd</b> Changes your login password
21	<b>Printenv</b> Displays value of a shell variable
22	<b>Ps</b> Displays the status of current processes

23	<b>Ps</b> Prints process status statistics
24	<b>quota -v</b> Displays disk usage and limits
25	<b>Reset</b> Resets terminal mode
26	<b>Script</b> Keeps script of terminal session
27	<b>Script</b> Saves the output of a command or process
28	<b>Setenv</b> Sets environment variables
30	<b>Sty</b> Sets terminal options
31	<b>Time</b> Helps time a command
32	<b>Top</b> Displays all system processes
33	<b>Tset</b> Sets terminal mode
34	<b>tty</b> Prints current terminal name
35	<b>umask</b> Show the permissions that are given to view files by default

36	<b>uname</b> Displays name of the current system
37	<b>uptime</b> Gets the system up time
38	<b>useradd</b> Creates a user account
39	<b>users</b> Prints names of logged in users
40	<b>vmstat</b> Reports virtual memory statistics
41	<b>w</b> Shows what logged in users are doing
42	<b>who</b> Lists logged in users

## **Linux Basic Commands**

Before we go on to the list of commands, you need to open the command line first. If you are still unsure about the command-line interface, check out this [CLI tutorial](#).

Although the steps may differ depending on the distribution that you're using, you can usually find the command line in the **Utilities** section.

Here is a list of basic Linux commands:

### **1. pwd command**

Use the **pwd** command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/). An example of an absolute path is **/home/username**.

### **2. cd command**

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.

Let's say you're in **/home/username/Documents** and you want to go to **Photos**, a subdirectory of **Documents**. To do so, simply type the following command: **cd Photos**.

Another scenario is if you want to switch to a completely new directory, for example, **/home/username/Movies**. In this case, you have to type **cd** followed by the directory's absolute path: **cd /home/username/Movies**.

There are some shortcuts to help you navigate quickly:

- **cd ..** (with two dots) to move one directory up
- **cd** to go straight to the home folder
- **cd-** (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

### 3. ls command

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type **ls** and then the directory's path. For example, enter **ls /home/username/Documents** to view the content of **Documents**.

There are variations you can use with the **ls** command:

- **ls -R** will list all the files in the sub-directories as well
- **ls -a** will show the hidden files
- **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

### 4. cat command

**cat** (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (stdout). To run this command, type **cat** followed by the file's name and its extension. For instance: **cat file.txt**.

Here are other ways to use the **cat** command:

- **cat > filename** creates a new file
- **cat filename1 filename2>filename3** joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, **cat filename | tr a-z A-Z >output.txt**

### 5. cp command

Use the **cp** command to copy files from the current directory to a different directory. For instance, the command **cp scenery.jpg /home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.

## 6. mv command

The primary use of the **mv** command is to move files, although it can also be used to rename files.

The arguments in **mv** are similar to the **cp** command. You need to type **mv**, the file's name, and the destination's directory. For example: **mv file.txt /home/username/Documents**.

To rename files, the Linux command is **mv oldname.ext newname.ext**

## 7. mkdir command

Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.

There are extra **mkdir** commands as well:

- To generate a new directory inside another directory, use this Linux basic command **mkdir Music/Newfile**
- use the **p** (parents) option to create a directory in between two existing directories. For example, **mkdir -p Music/2020/Newfile** will create the new “2020” file.

## 8. rmdir command

If you need to delete a directory, use the **rmdir** command. However, **rmdir** only allows you to delete empty directories.

## 9. rm command

The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to **rmdir** — use **rm -r**.

**Note:** Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.

## 10. touch command

The **touch** command allows you to create a blank new file through the Linux command line. As an example, enter **touch /home/username/Documents/Web.html** to create an HTML file entitled **Web** under the **Documents** directory.

## 11. locate command

You can use this command to **locate** a file, just like the search command in Windows. What's more, using the **-i** argument along with this command will make it case-insensitive, so you can search for a file even if you don't remember its exact name.

To search for a file that contains two or more words, use an asterisk (\*). For example, **locate -i school\*note** command will search for any file that contains the word "school" and "note", whether it is uppercase or lowercase.

## 12. find command

Similar to the **locate** command, using **find** also searches for files and directories. The difference is, you use the **find** command to locate files within a given directory.

As an example, **find /home/ -name notes.txt** command will search for a file called **notes.txt** within the home directory and its subdirectories.

Other variations when using the **find** are:

- To find files in the current directory use, **find . -name notes.txt**
- To look for directories use, **/ -type d -name notes.txt**

## 13. grep command

Another basic Linux command that is undoubtedly helpful for everyday use is **grep**. It lets you search through all the text in a given file.

To illustrate, **grep blue notepad.txt** will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully.

## 14. sudo command

Short for "**SuperUser Do**", this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

## 15. df command

Use **df** command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type **df -m**.

## 16. du command

If you want to check how much space a file or a directory takes, the **du** (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the **-h** argument to the command line.

## 17. head command

The **head** command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type **head -n 5 filename.ext**.

## 18. tail command

This one has a similar function to the head command, but instead of showing the first lines, the **tail** command will display the last ten lines of a text file. For example, **tail -n filename.ext**.

## 19. diff command

Short for difference, the **diff** command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

The simplest form of this command is **diff file1.ext file2.ext**

## 20. tar command

The **tar** command is the most used command to archive multiple files into a **tarball** — a common Linux file format that is similar to zip format, with compression being optional.

This command is quite complex with a long list of functions such as adding new files into an existing archive, listing the content of an archive, extracting the content from an archive, and many more. Check out some [practical examples](#) to know more about other functions.

## 21. chmod command

**chmod** is another Linux command, used to change the read, write, and execute permissions of files and directories. As this command is rather complicated, you can read [the full tutorial](#) in order to execute it properly.

## 22. chown command

In Linux, all files are owned by a specific user. The **chown** command enables you to change or transfer the ownership of a file to the specified username. For instance, **chown linuxuser2 file.ext** will make **linuxuser2** as the owner of the **file.ext**.

## 23. jobs command

**jobs** command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

## 24. kill command

If you have an unresponsive program, you can terminate it manually by using the **kill** command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of [sixty-four signals](#) that you can use, but people usually only use two signals:

- **SIGTERM (15)** — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.
- **SIGKILL (9)** — forces programs to stop immediately. Unsaved progress will be lost.

Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to **kill**. If you don't know the PID, simply run the command **ps ux**.

After knowing what signal you want to use and the PID of the program, enter the following syntax:

**kill [signal option] PID.**

## 25. ping command

Use the **ping** command to check your connectivity status to a server. For example, by simply entering **ping google.com**, the command will check whether you're able to connect to Google and also measure the response time.

## 26. wget command

The Linux command line is super useful — you can even download files from the internet with the help of the **wget** command. To do so, simply type **wget** followed by the download link.

## 27. uname command

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

## 28. top command

As a terminal equivalent to Task Manager in Windows, the **top** command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

## 29. history command

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running **history** command is particularly useful if you want to review the commands you've entered before.



### 30. man command

Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the **man** command. For instance, entering **man tail** will show the manual instruction of the tail command.

### 31. echo command

This command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called name.txt, you would type **echo Hello, my name is John >> name.txt**

### 32. zip, unzip command

Use the **zip** command to compress your files into a zip archive, and use the **unzip** command to extract the zipped files from a zip archive.

### 33. hostname command

If you want to know the name of your host/network simply type **hostname**. Adding a **-I** to the end will display the IP address of your network.

### 34. useradd, userdel command

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. **useradd** is used to create a new user, while **passwd** is adding a password to that user's account. To add a new person named John type, **useradd John** and then to add his password type, **passwd 123456789**.

To remove a user is very similar to adding a new user. To delete the users account type, **userdel UserName**

### Bonus Tips and Tricks

Use the **clear** command to clean out the terminal if it is getting cluttered with too many past commands.

Try the **TAB** button to autofill what you are typing. For example, if you need to type Documents, begin to type a command (let's go with **cd Docu**, then hit the TAB key) and the terminal will fill in the rest, showing you **cd Documents**.

**Ctrl+C** and **Ctrl+Z** are used to stop any command that is currently working. Ctrl+C will stop and terminate the command, while Ctrl+Z will simply pause the command.

If you accidental freeze your terminal by using **Ctrl+S**, simply undo this with the unfreeze **Ctrl+Q**.

**Ctrl+A** moves you to the beginning of the line while **Ctrl+E** moves you to the end.

You can run multiple commands in one single command by using the “;” to separate them. For example **Command1; Command2; Command3**. Or use **&&** if you only want the next command to run when the first one is successful.

## To Sum Up

Basic Linux commands help users execute tasks easily and effectively. It might take a while to remember some of the basic commands, but nothing is impossible with lots of practice.

In the end, knowing and mastering these basic Linux commands will be undoubtedly beneficial for you. Good luck!

## How to Install Linux?

---

[Linux is an](#) open source and free operating system to install which allows anyone with programming knowledge to modify and create its own operating system as per their requirements. Over many years, it has become more user-friendly and supports a lot of features such as

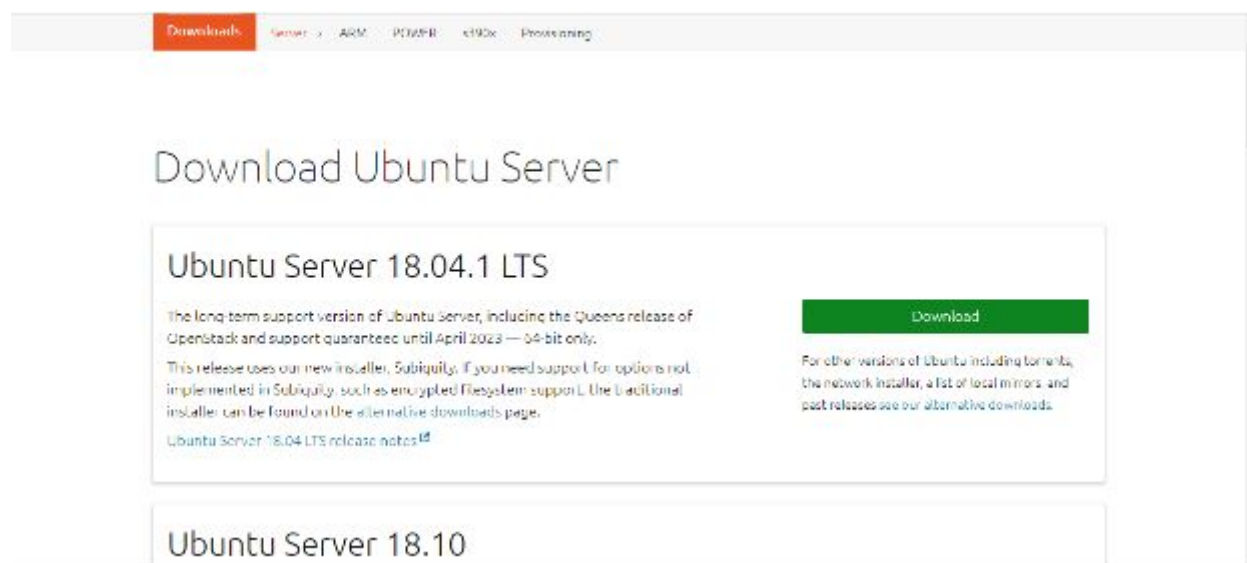
1. Reliable when used with servers
2. No need of antivirus
3. A Linux server can run nonstop with the boot for many years.

It has many distributions such as Ubuntu, [Fedora](#), Redhat, Debian but all run on top of Linux server itself. Installation of every distribution is similar, thus we are explaining Ubuntu here.

So let's get started using this wonderful operating system by any of the following methods.

### A. Install Linux Using CD-ROM or USB Stick

Download .iso or the ISO files on a computer from the internet and store it in the CD-ROM or USB stick after making it bootable using Pen Drive Linux and UNetBootin



## 1. Boot into the USB Stick

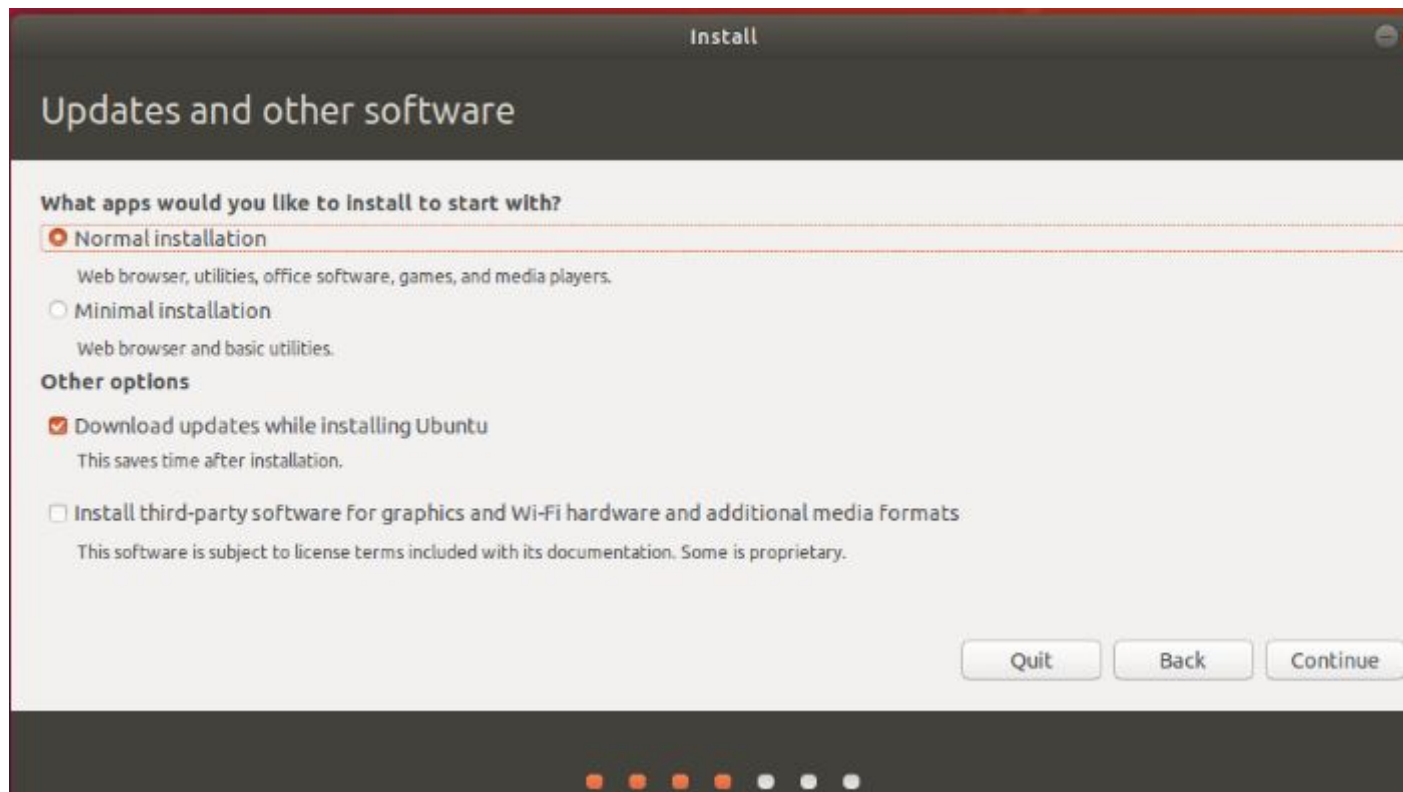
You need to restart your computer after attaching CD –ROM or pen drive into the computer. Press enter at the time of boot, here select the CD-ROM or pen drive option to start the further boot process. Try for a manual boot setting by holding F12 key to start the boot process. This will allow you to select from various boot options before starting the system. All the options either it is USB or CD ROM or number of operating systems you will get a list from which you need to select one.

**Note:-**

You will see a new screen when your computer boots up called “GNU GRUB”, a boot loader that handles installations for Linux. This screen will only appear in case there is more than one operating system.



- Set the keyboard layout.
- Now you will be asked What apps would you like to install to start with Linux? The two options are ‘Normal installation’ and ‘Minimal installation’.



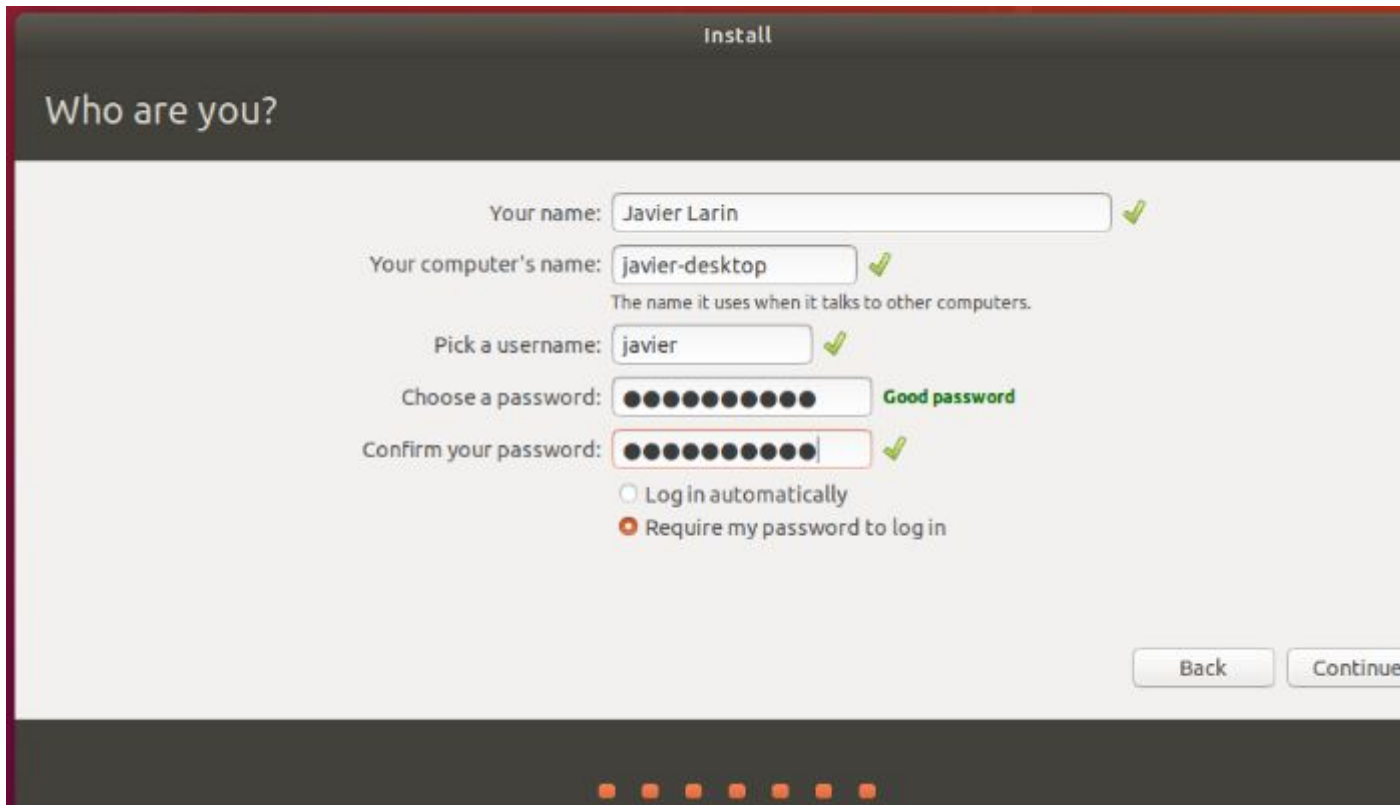
## 2. Derive Selection

Select the drive for installation of OS to be completed. Select “Erase Disk and [install Ubuntu](#)” in case you want to replace the existing OS otherwise select “Something else” option and click INSTALL NOW.



### 3. Start Installation

- A small panel will ask for confirmation. Click Continue in case you don't want to change any information provided. Select your location on the map and install Linux.
- Provide the login details.



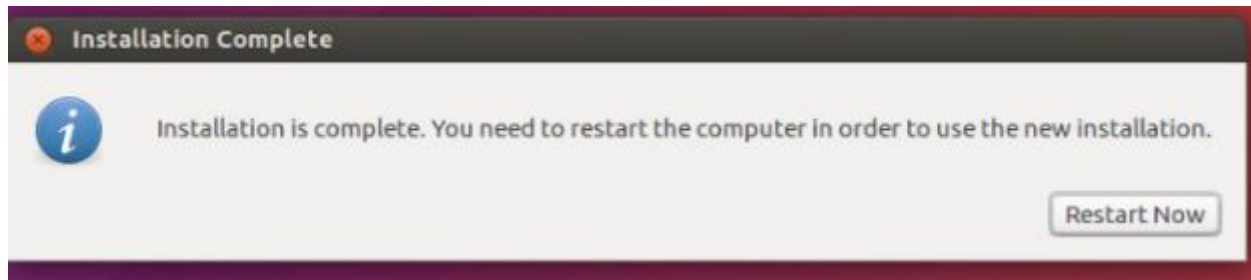
The screenshot shows the 'Install' window with the title 'Who are you?'. It contains several input fields for user information, each with a green checkmark indicating successful validation:

- Your name:** Javier Larin
- Your computer's name:** javier-desktop (with a subtext: 'The name it uses when it talks to other computers.')
- Pick a username:** javier
- Choose a password:** (masked with dots) with a green label 'Good password' to its right.
- Confirm your password:** (masked with dots) with a green checkmark to its right.

At the bottom, there are two radio buttons: 'Log in automatically' (unselected) and 'Require my password to log in' (selected). In the bottom right corner, there are 'Back' and 'Continue' buttons. At the very bottom of the window, there is a progress bar with seven orange squares.

### 4. Complete the installation process

After the installation is complete you will see a prompt to restart the computer.



You can also download drivers of your choice through the System Settings menu. Just follow these steps:

Additional Drivers > select the graphics driver from the list.

Many useful drivers will be available in the list, such as Wi-Fi drivers.

There are many other options also available to use and install Linux

## B. Install Linux Using Virtual Box VMWARE

In this way, nothing will affect your Windows operating system.

### *What Are Requirements?*

- Good internet connection
- At least 4GB RAM
- At least 12GB of free space

### *Steps:*

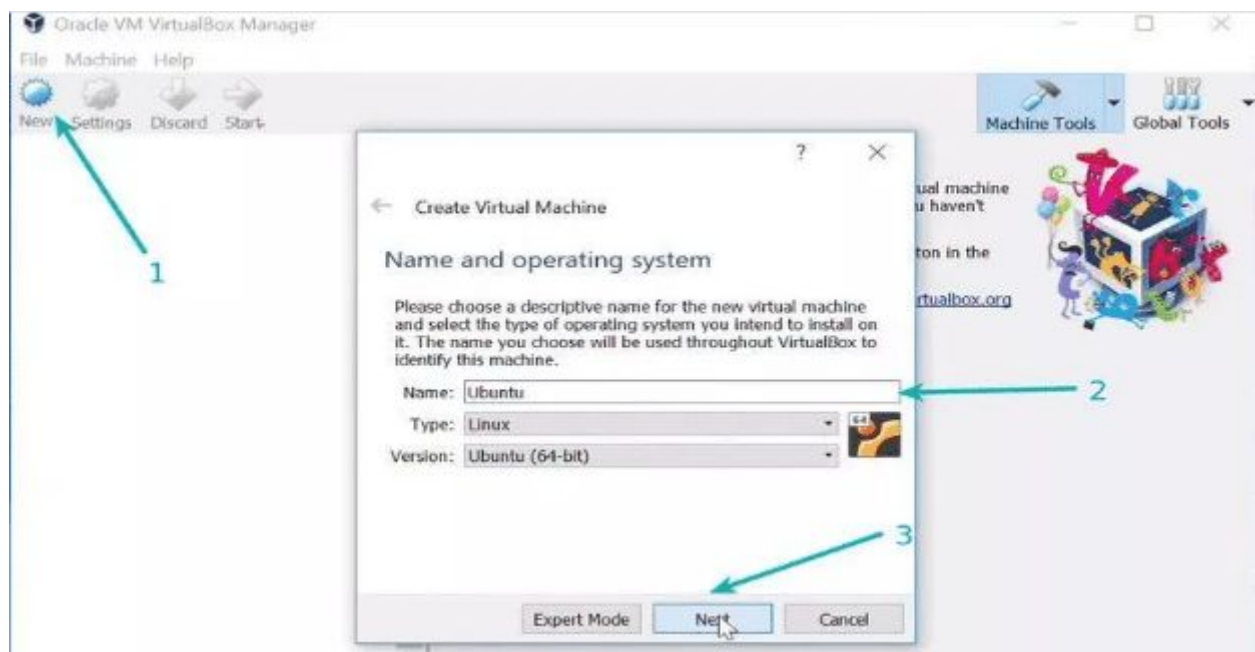
1. Download the VIRTUAL BOX from original ORACLE VIRTUAL BOX site. You can refer below link

<https://www.virtualbox.org/>



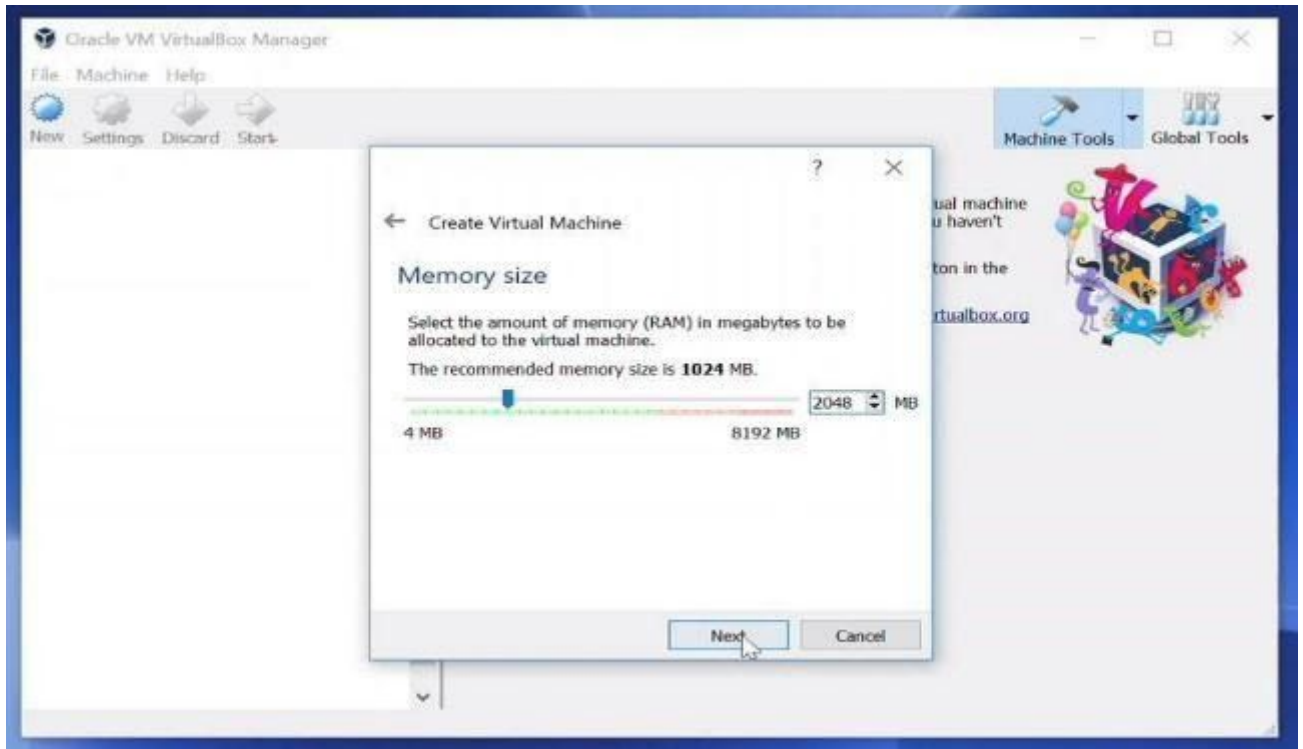
## 2. Install Linux Using Virtual Box

Use the .iso file or ISO file that can be downloaded from the internet and start the virtual box.

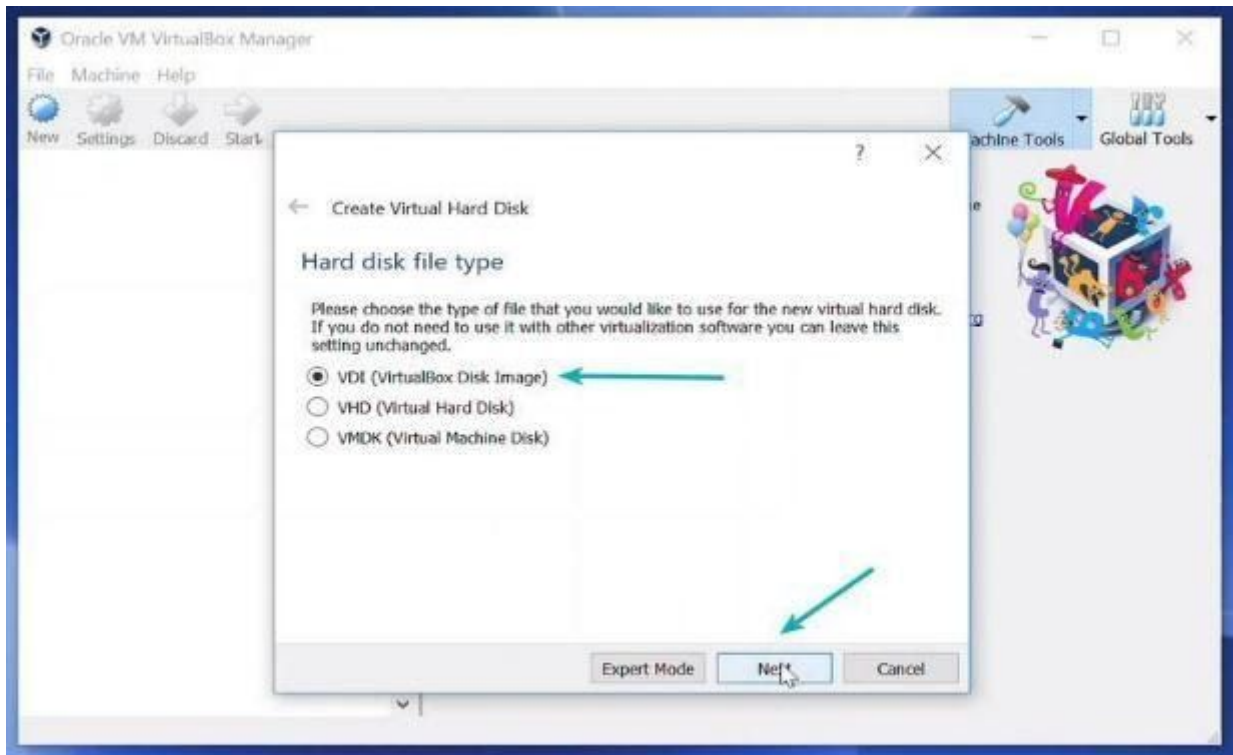




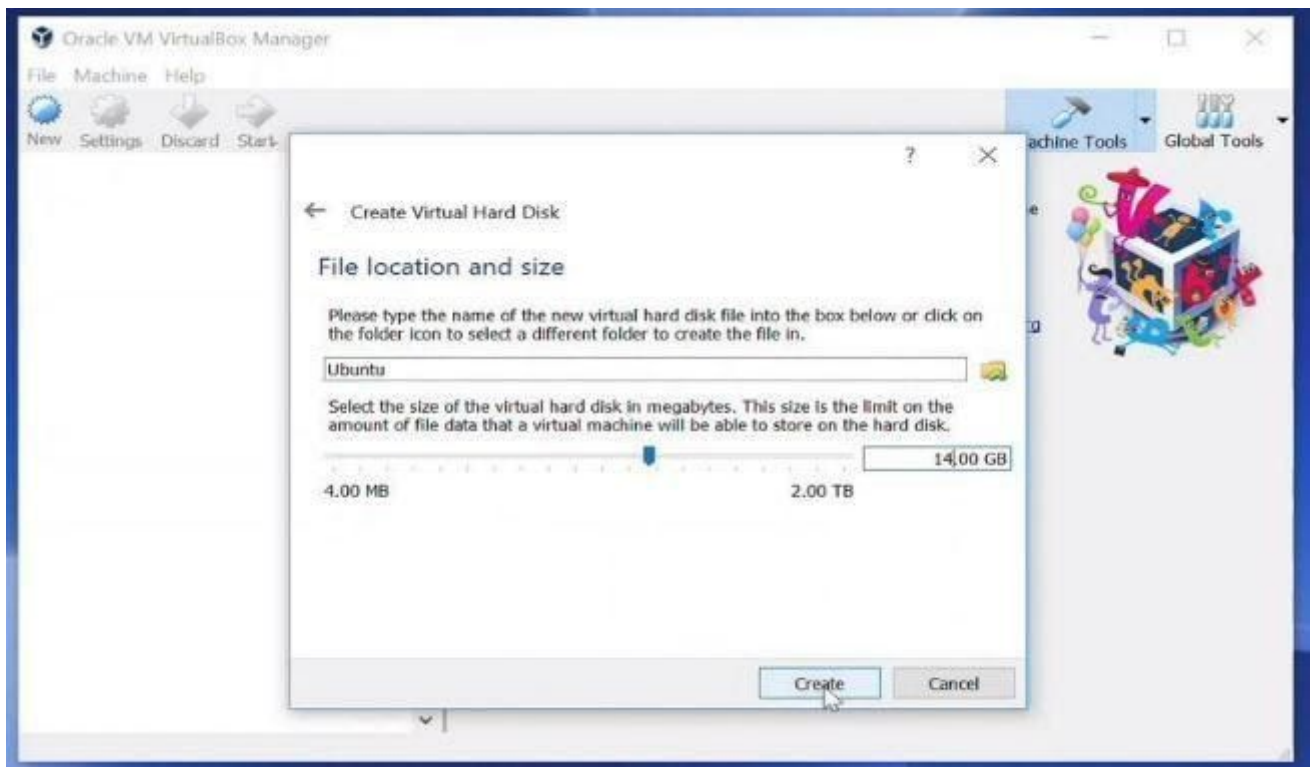
Here we need to allocate RAM to virtual OS. It should be 2 GB as per minimum requirement.



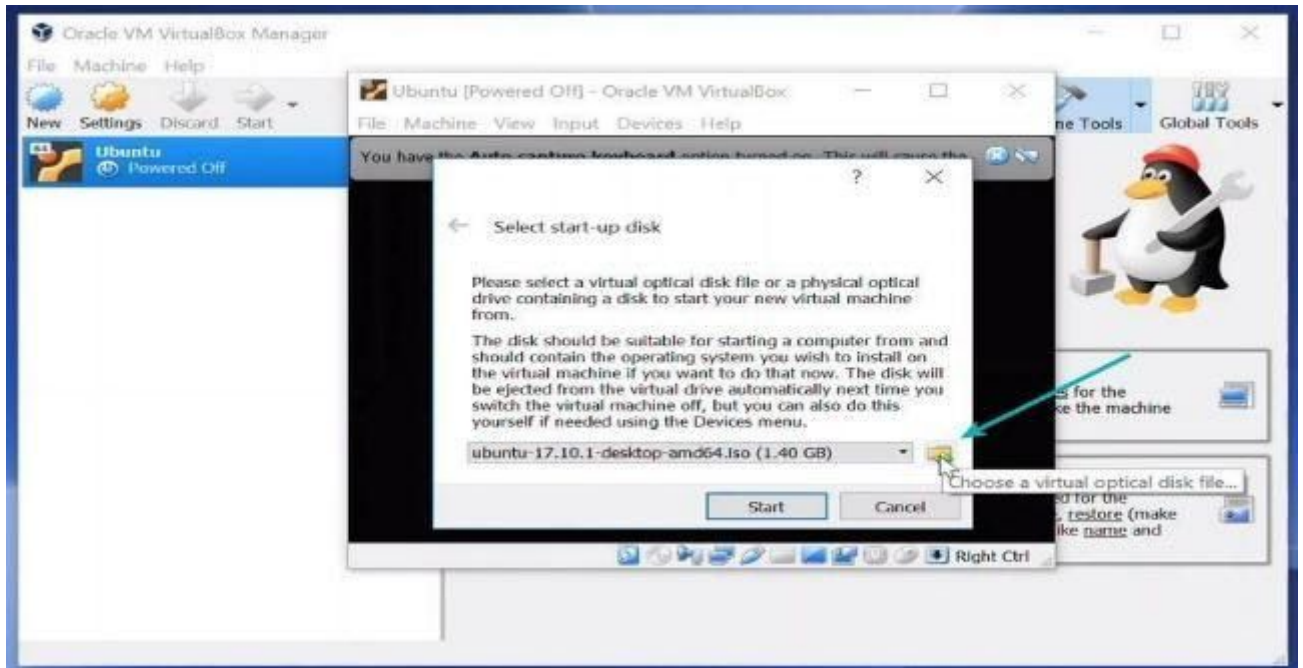
- Choose an option under Create a virtual disk.



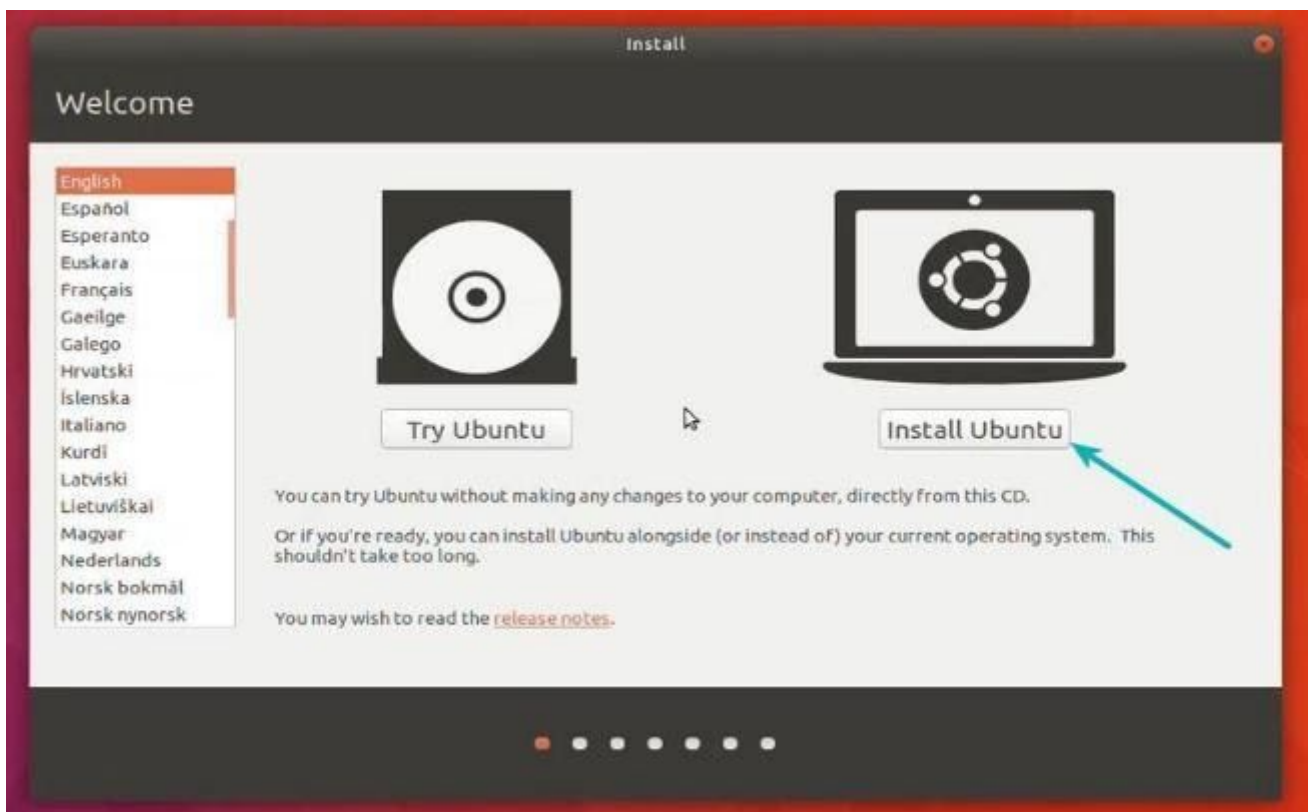
- Choose a type of storage on physical hard disk. And choose the disk size(min 12 GB as per requirement)

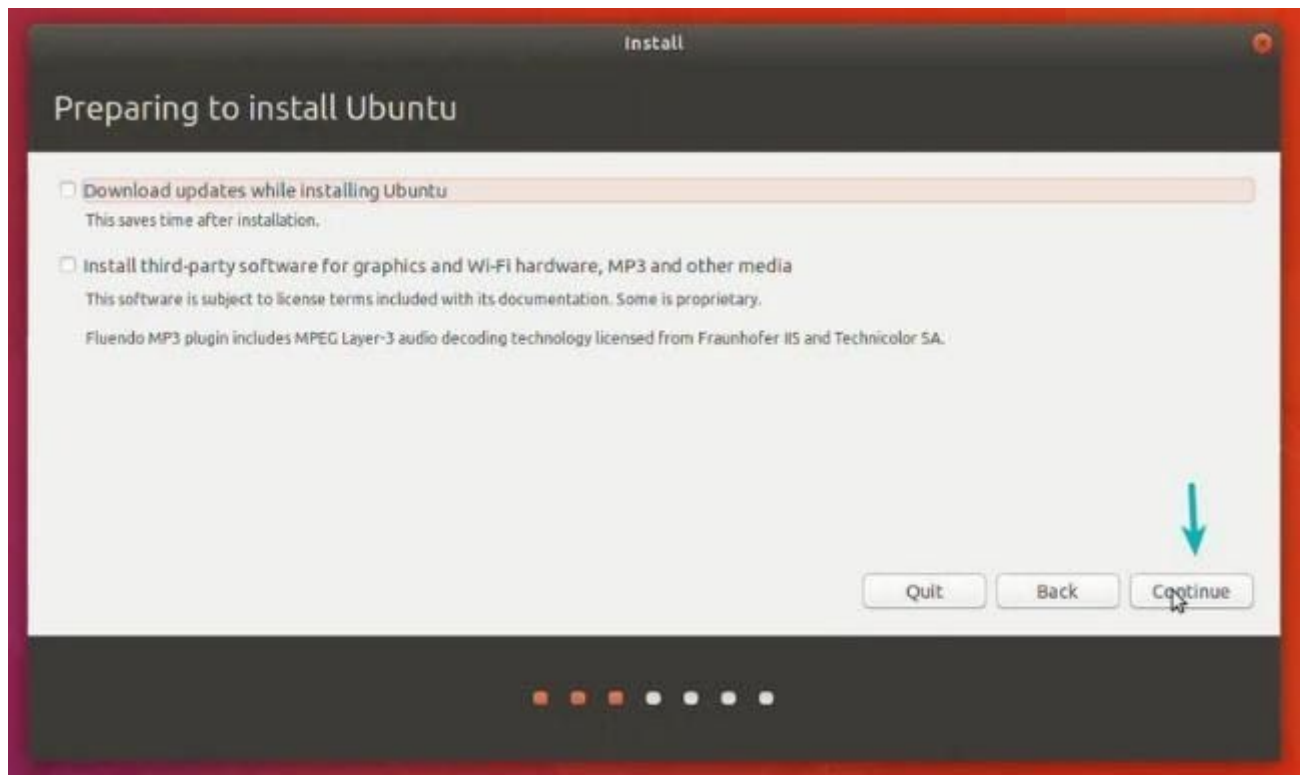


- Click on create option and then click on the START button to start the virtual box and browse to the location of the .iso file of the OS.

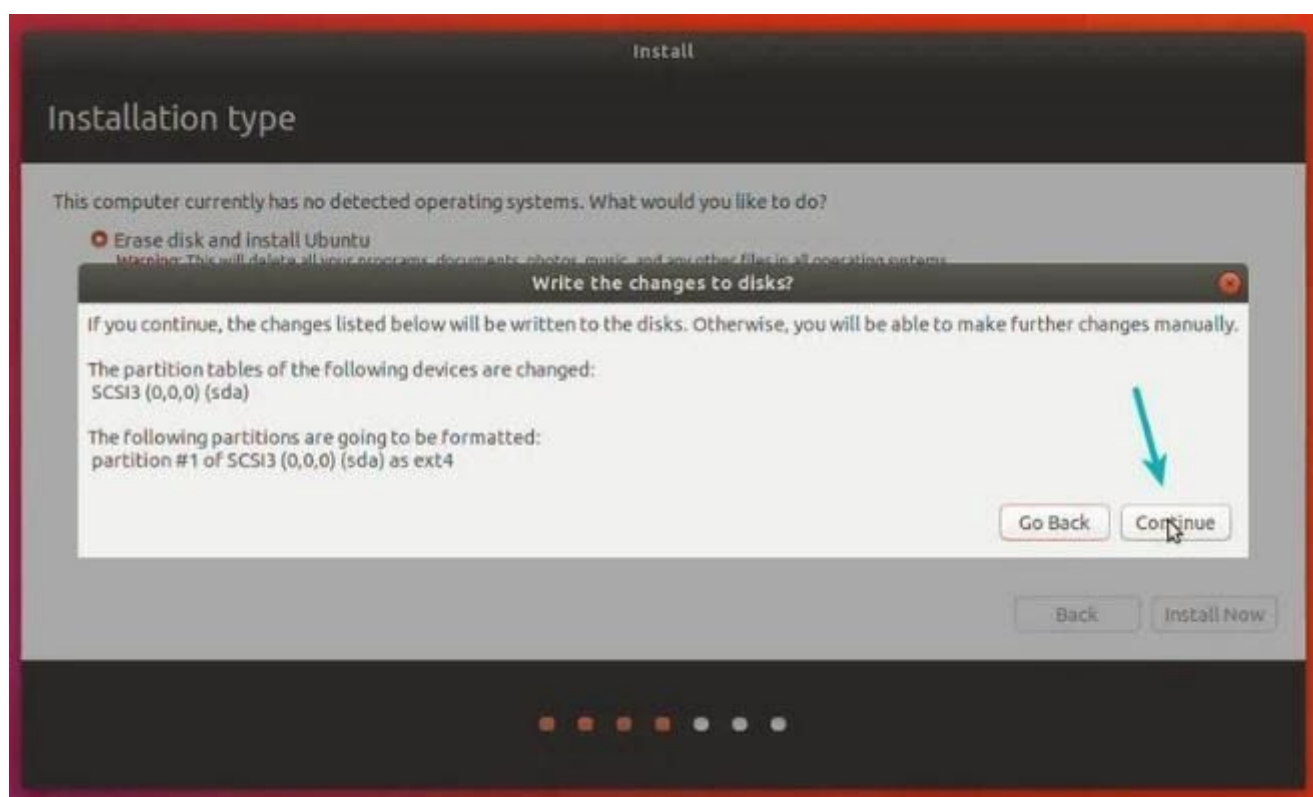
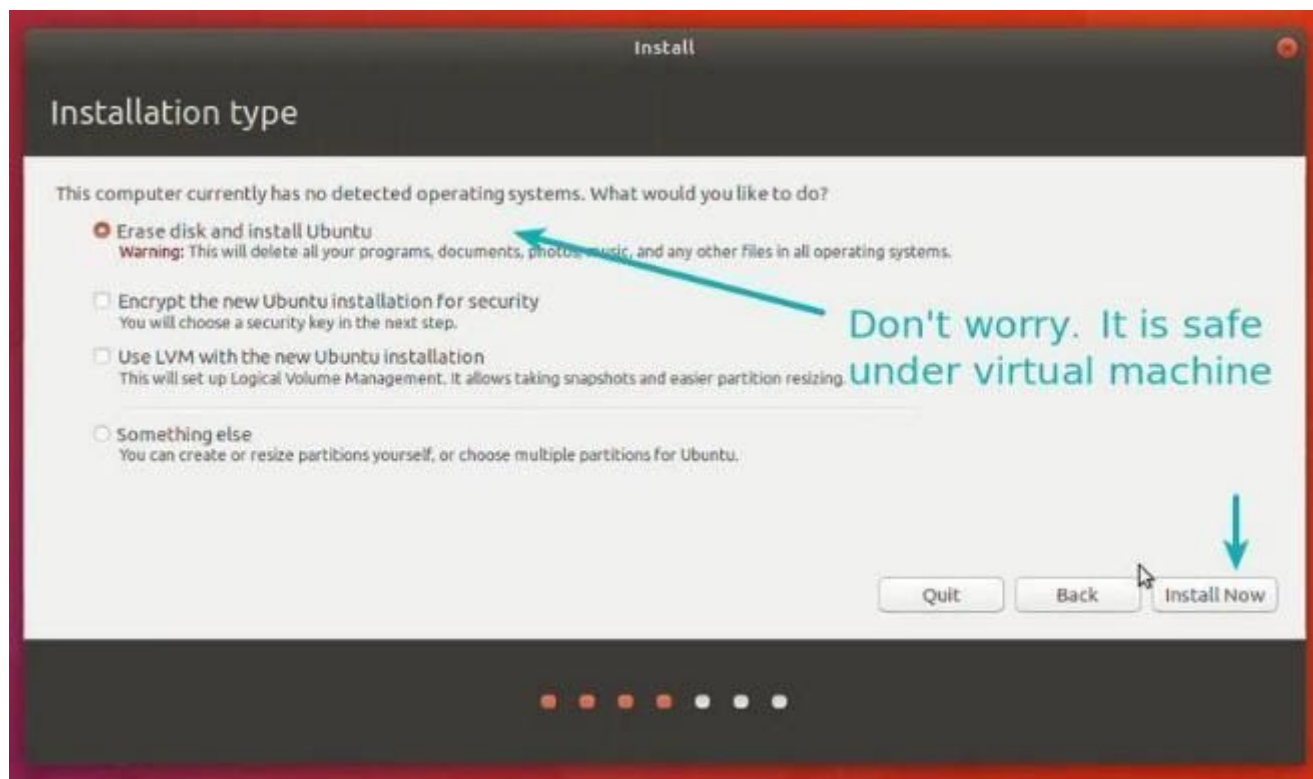


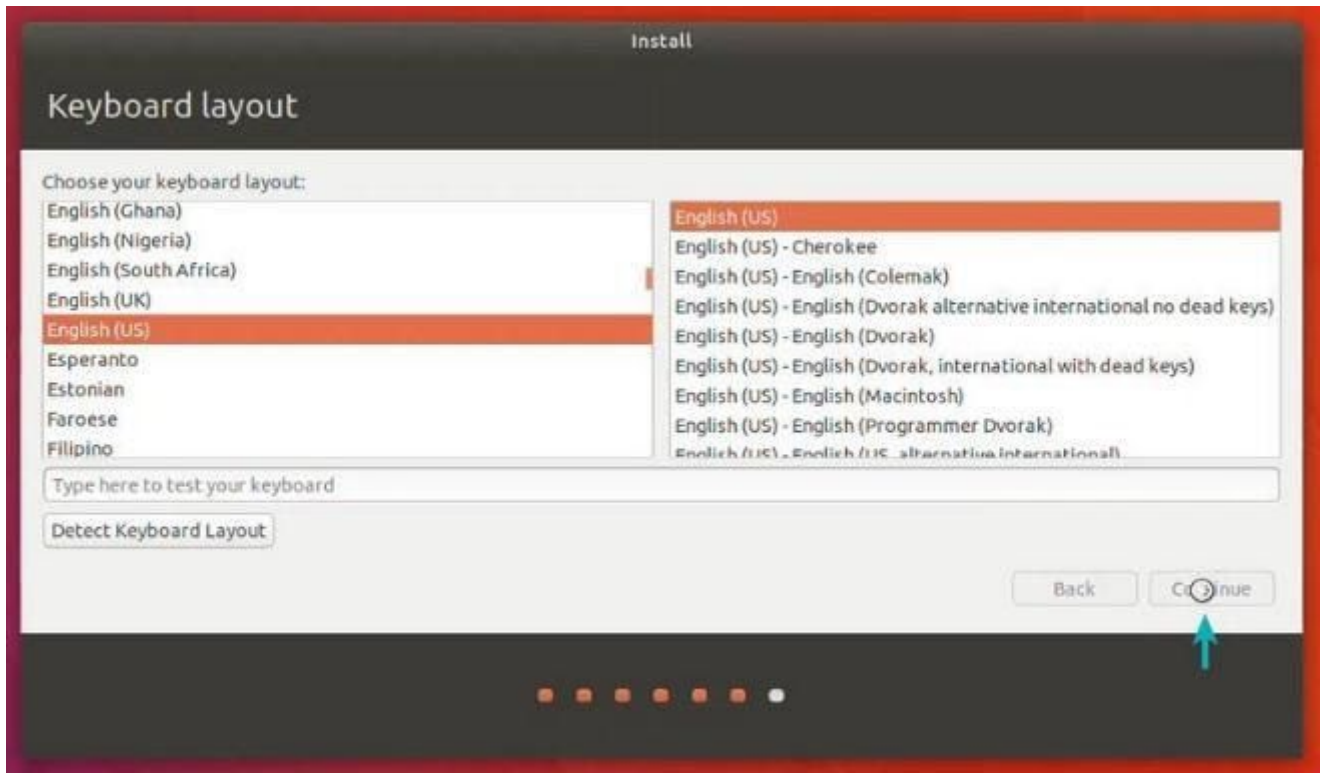
- Now Linux OS will start, Click on install option.



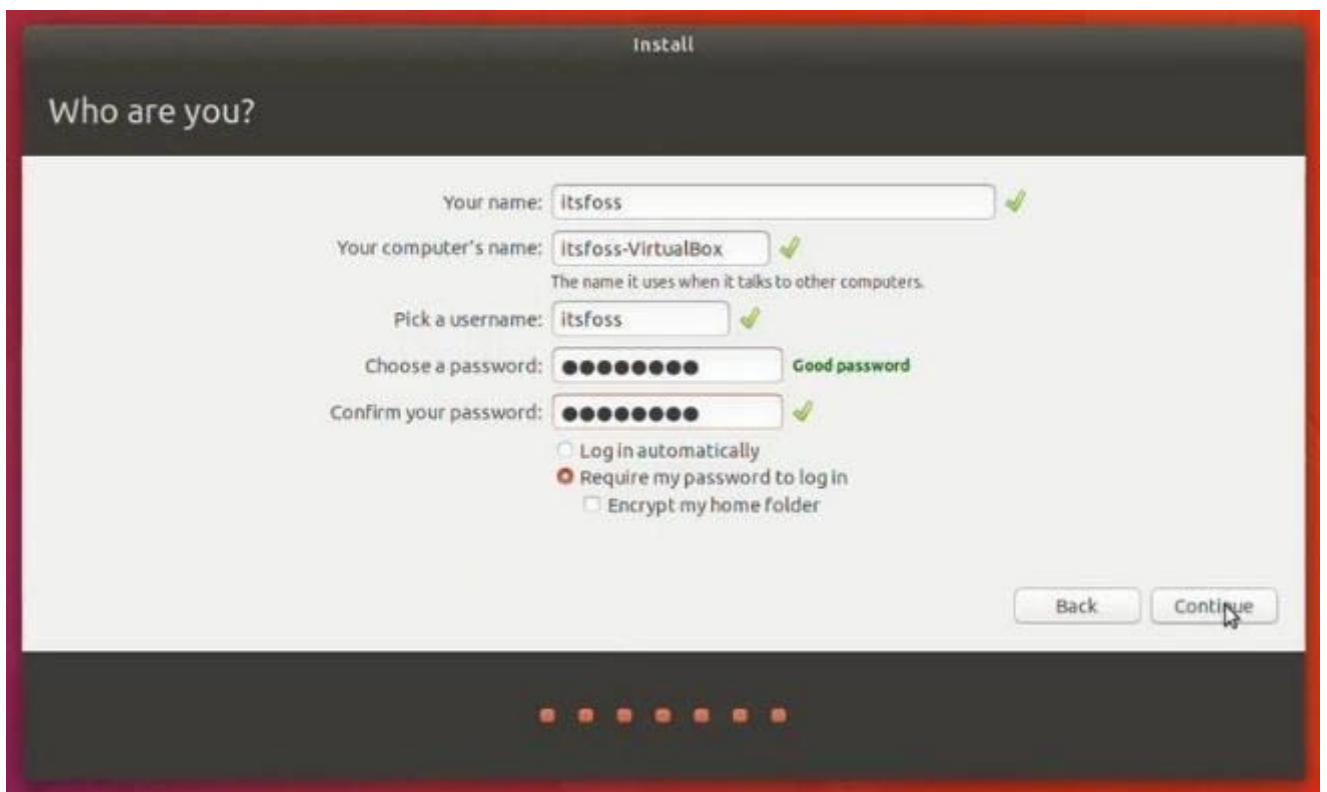


- Select the drive for completing the OS installation. Select “Erase Disk and install Ubuntu” in case you want to replace the existing OS otherwise select “Something else” option and click INSTALL NOW.

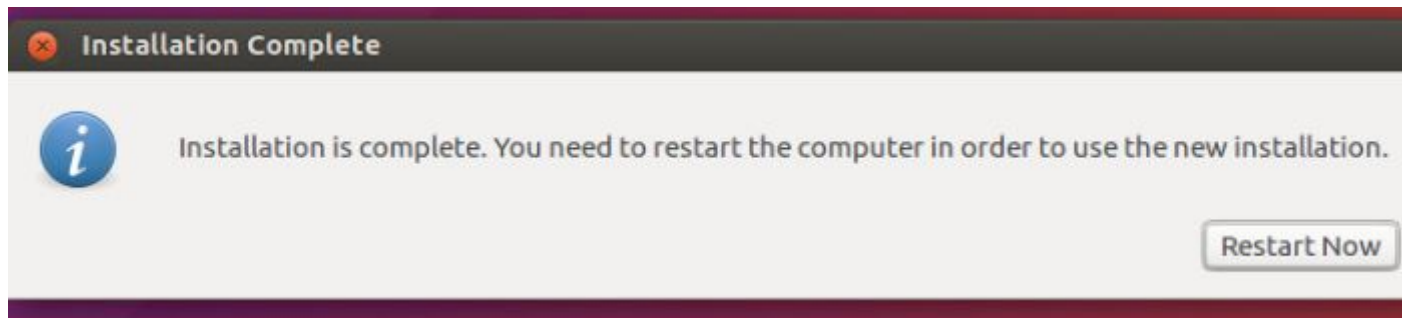




- Click on Continue.
- Choose a username and password.



You are almost done. It should take 10–15 minutes to complete the installation. Once the installation finishes, restart the system.



**NOTE:** In case of any issue close and again start the virtual box.

The Linux operating systems now offer millions of programs/applications to choose from, most of them free to install! Linux is also the OS of choice for Server environments due to its stability and reliability (Mega-companies like Amazon, Facebook, and Google use [Linux](#) for their Servers). It proves to be a good choice for everyone.