

DEPARTMENT OF COMPUTER APPLICATION
TKM COLLEGE OF ENGINEERING
KOLLAM – 691005



**20MCA136 - NETWORKING & SYSTEM ADMINISTRATION
LAB**
PRACTICAL RECORD BOOK
Second Semester MCA
2020-2021

Submitted by:

NAME : ROBIN MONACHAN

ROLL NO : MCA232

**DEPARTMENT OF COMPUTER APPLICATION
TKM COLLEGE OF ENGINEERING
KOLLAM – 691005**



CERTIFICATE

This is a bonafide record of the work done by ROBIN MONACHAN in the Second Semester in NETWORKING & SYSTEM ADMINISTRATION LAB Course (20MCA136) towards the partial fulfillment of the degree of Master of Computer Applications during the academic year 2020-2021.

Staff Member in-charge

Examiner

.....

.....

INDEX

	Contents	Date	Page No.
	Cycle 1		
Exp 1	Familiarisation to computer hardware	28/04/21	1
Exp 2	Specification of desktop and server computers	28/04/21	8
Exp 3	Installation of linux on virtual box in windows 10	30/04/21	12
Exp 4	Installation of windows on virtual box in ubuntu	05/05/21	15
Exp 5	Installation of windows operating system on a bare machine	21/05/21	17
Exp 6	Installation of linux over windows operating system	02/06/21	21
Exp 7	Installation of server system	07/05/21	30
	Cycle 2		
Exp 8	Study on command line text editors	02/06/21	33
Exp 9	Basic Linux Commands	02/06/21	35
Exp 10	Familiarisation with Vi Editor	04/06/21	55
Exp 11	Familiarisation with EMACS editor	09/06/21	64
Exp 12	Linux File System	16/06/21	73
	Cycle 3		
Exp 13	Familiarisation to linux shell and shell scripting	18/06/21	87
Exp 14	Linux shell scripting problems	06/08/21	98
	Cycle 4		
Exp 15	Installation of LAMP	13/08/21	109
Exp 16	Installation of Laravel	1/09/21	113
	Cycle 5		
Exp 17	Familiarisation to command Line tool for networking	03/09/21	117
Exp 18	Familiarisation of Static and Dynamic IP	03/09/21	126
Exp 19	Concept of subnets and CIDR address scheme	05/09/21	130
Exp 20	Concept of subnet mask	05/09/21	133
Exp 21	Setting up of firewall on LAN	07/09/21	134
Exp 22	Wireshark and TCP Dump	08/09/21	135
Exp 23	Analyse Packets using wireshark	08/09/21	151
	Cycle 6		
Exp 24	Familiarisation to hypervisors and Virtual machines	15/09/21	156
Exp 25	Familiarisation to Xen or KVM	15/09/21	158
Exp 26	Installation of software from source code	19/09/21	164
Exp 27	Deploy Linux virtual machine using Ansible Playbook	19/09/21	168

EXPERIMENT NO: 1

FAMILIARISATION TO COMPUTER HARDWARE

Computer hardware includes the physical parts of a computer, such as the central processing unit (CPU), monitor, mouse, keyboard, computer data storage, graphics card, sound card, speakers and motherboard.

By contrast, software is the set of instructions that can be stored and run by hardware. Hardware is so termed because it is "hard" or rigid with respect to changes, whereas software is "soft" because it is easy to change.

Hardware is typically directed by the software to execute any command or instruction. A combination of hardware and software forms a usable computing system, although other systems exist with only hardware.

Mother Board

The motherboard is the body or mainframe of the computer, through which all other components interface. It is the central circuit board making up a complex electronic system. A motherboard provides the electrical connections by which the other components of the system communicate. The mother board includes many components such as: central processing unit (CPU), random access memory (RAM), firmware, and internal and external buses.



Processor

The processor is the most important component placed on the motherboard, present in the computer case as a CPU (Central Processing Unit). The processor acts as the brain of the entire operation of the computer system and it is the 4th generation of computer.

The processor unit allows the computer to perform different tasks like to process the data, control the operation of all the computer's devices, and most importantly performing logical and mathematical operations.

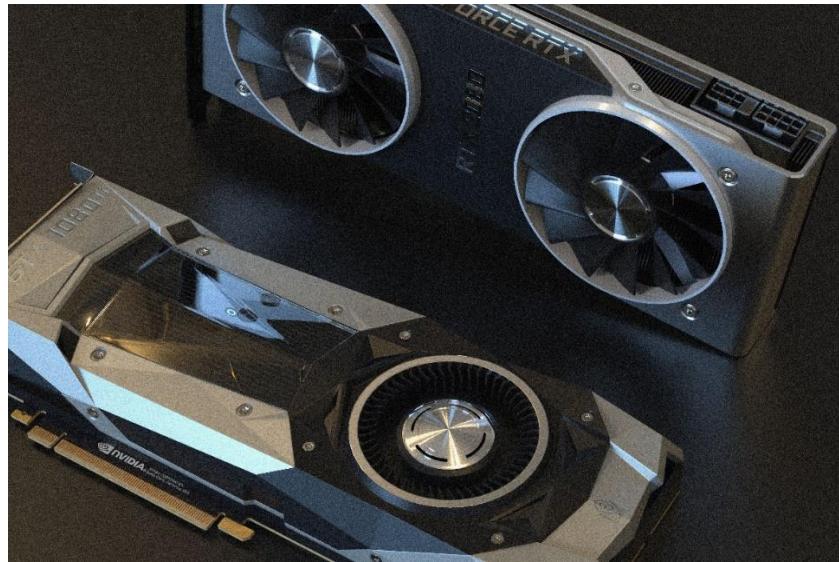
And other actions like controlling the flow of information within the PC, managing and controlling the RAM and ROM memory, and performing basic operations on the computer's data.

In simple words, it is just processes everything that happens on the computer and executes all the actions. The faster the processor a computer has, the faster the computer will work.



GPU

A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing. Their highly parallel structure makes them more efficient than general-purpose central processing units (CPUs) for algorithms that process large blocks of data in parallel. In a personal computer, a GPU can be present on a video card or embedded on the motherboard. In certain CPUs, they are embedded on the CPU die.^[1]



SMPS

A switched-mode power supply (SMPS) is an electronic circuit that converts power using switching devices that are turned on and off at high frequencies, and storage components such as inductors or capacitors to supply power when the switching device is in its non-conduction state.

Switching power supplies have high efficiency and are widely used in a variety of electronic equipment, including computers and other sensitive equipment requiring stable and efficient power supply.

A switched-mode power supply is also known as a switch-mode power supply or switching-mode power supply.



Random Access Memory

Random access memory (RAM) is fast-access memory that is cleared when the computer is power-down. RAM attaches directly to the motherboard, and is used to store programs that are currently running. RAM is a set of integrated circuits that allow the stored data to be accessed in any order (why it is called random). There are many different types of RAM. Distinctions between these different types include: writable vs. read-only, static vs. dynamic, volatile vs. non-volatile, etc.



ROM

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture. A ROM stores such instructions that are required to start a computer. This operation is referred to as bootstrap. ROM chips are not only used in the computer but also in other electronic items like washing machine and microwave oven.



Hard Disk Drive

A hard disk drive (HDD) is a non-volatile storage device which stores digitally encoded data on rapidly rotating platters with magnetic surfaces. Just about every new computer comes with a hard disk these days unless it comes with a new solid-state drive. Typical desktop hard disk drives store between 120 and 400GB, rotate at 7,200 rpm, and have a media transfer rate of 1 Gbit/s or higher. Hard disk drives are accessed over one of a number of bus types, including parallel ATA(also called IDE), Serial ATA (SATA), SCSI, Serial Attached SCSI, and Fibre Channel.



SSD

A solid-state drive (SSD) is a solid-state storage device that uses integrated circuit assemblies to store data persistently, typically using flash memory, and functioning as secondary storage in the hierarchy of computer storage. It is also sometimes called a solid-state device or a solid-state disk,^[1] even though SSDs lack the physical spinning disks and movable read–write heads used in hard disk drives (HDDs) and floppy disks.



Monitor

The computer monitor is a significant part, without it the user cannot function the computer. The screen of the monitor allows the user to interact with the computer. The monitor screen is for visual display of all types of information provided by the computer.

The **main function of the monitor is obviously visual**. As it acts as an interface between the CPU and the user. It doesn't matter how powerful or fast your computer is, without a monitor display, the computer is incomplete or even useless.

The monitor is designed to display all kinds of information like image, video, symbolic, graphical, etc, as Soft Copy on its screen. A cable is connected with a video adapter that is set up with the computer's motherboard to display the given data.

Through computer monitors, we can carry out, or view all the important content, review stored information, and a lot of tasks.



Keyboard

The technology of computer keyboards includes many elements. Among the more important of these is the switch technology that they use. Computer alphanumeric keyboards typically have 80 to 110 durable switches, generally one for each key. The choice of switch technology affects key response (the positive feedback that a key has been pressed) and pre-travel (the distance needed to push the key to enter a character reliably). Virtual keyboards on touch screens have no physical switches and provide audio and haptic feedback instead. Some newer keyboard models use hybrids of various technologies to achieve greater cost savings or better ergonomics.



Mouse

A computer mouse (plural mice, rarely mouses) is a hand-held pointing device that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of a pointer on a display, which allows a smooth control of the graphical user interface of a computer.



EXPERIMENT NO: 2

SPECIFICATION OF DESKTOP AND SERVER COMPUTERS

- **Processor:** 10th or 11th Gen Intel Core i5, i7 or i9 Processor, or Apple M1 Processor (CPU)
- **Operating System:** Microsoft Windows 10 Home, Pro, Enterprise or Education version
- **Memory (RAM):** 8-16 GB of RAM
- **Warranty:** 3 year on-site (better) or depot warranty, accidental damage service highly recommended.
- **Storage:** All the files, programs and games on your PC are kept on its storage or hard drive, which you can think of as the computer's "long-term memory". The more storage available, the more you can keep on there. Desktop PCs usually have lots of storage, usually starting at 500GB to 1TB, which is plenty for the average user. Storage is measured in gigabytes (GB) and terabytes (TB), with 1TB equal to 1,000 GB
- **Monitor:** A desktop monitor is the screen you use to operate your PC. When buying a monitor. Much like a TV, the main things to consider are screen size, resolution and budget
- **Mouse:** The mouse has been the main way of operating PCs for decades, offering a simple and effective way to use the operating system, as well as for typing, browsing and gaming.
- **Keyboard:** The keyboard has been the main way of operating PCs for decades, offering a simple and effective way to use the operating system, as well as for typing, browsing and gaming.

Server type	Purpose	Clients
Application server	Hosts web apps (computer programs that run inside a web browser) allowing users in the network to run and use them, without having to install a copy on their own computers. Unlike	Computers with a web browser

	what the name might imply, these servers need not be part of the World Wide Web; any local network would do.	
Catalog server	Maintains an index or table of contents of information that can be found across a large distributed network, such as computers, users, files shared on file servers, and web apps. Directory servers and name servers are examples of catalog servers.	Any computer program that needs to find something on the network, such a Domain member attempting to log in, an email client looking for an email address, or a user looking for a file
Communications server	Maintains an environment needed for one communication endpoint (user or devices) to find other endpoints and communicate with them. It may or may not include a directory of communication endpoints and a presence detection service, depending on the openness and security parameters of the network	Communication endpoints (users or devices)
Computing server	Shares vast amounts of computing resources, especially CPU and random-access memory, over a network.	Any computer program that needs more CPU power and RAM than a personal computer can probably afford. The client must be a networked computer; otherwise, there would be no client-server model.
Database server	Maintains and shares any form of database (organized collections of data with predefined properties that	Spreadsheets, accounting software, asset management software or virtually any computer program that

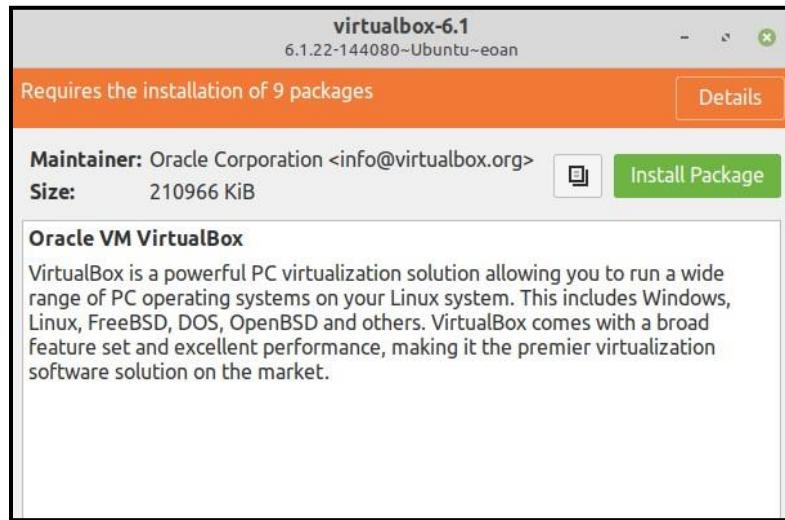
	may be displayed in a table) over a network.	consumes well-organized data, especially in large volumes
Fax server	Shares one or more fax machines over a network, thus eliminating the hassle of physical access	Any fax sender or recipient
File server	Shares files and folders, storage space to hold files and folders, or both, over a network	Networked computers are the intended clients, even though local programs can be clients
Game server	Enables several computers or gaming devices to play multiplayer video games	Personal computers or gaming consoles
Mail server	Makes email communication possible in the same way that a post office makes snail mail communication possible	Senders and recipients of email
Media server	Shares digital video or digital audio over a network through media streaming (transmitting content in a way that portions received can be watched or listened to as they arrive, as opposed to downloading an entire file and then using it)	User-attended personal computers equipped with a monitor and a speaker
Print server	Shares one or more printers over a network, thus eliminating the hassle of physical access	Computers in need of printing something

Sound server	Enables computer programs to play and record sound, individually or cooperatively	Computer programs of the same computer and network clients.
Proxy server	Acts as an intermediary between a client and a server, accepting incoming traffic from the client and sending it to the server. Reasons for doing so include content control and filtering, improving traffic performance, preventing unauthorized network access or simply routing the traffic over a large and complex network.	Any networked computer
Virtual server	Shares hardware and software resources with other virtual servers. It exists only as defined within specialized software called hypervisor. The hypervisor presents virtual hardware to the server as if it were real physical hardware. ^[7] Server virtualization allows for a more efficient infrastructure. ^[8]	Any networked computer
Web server	Hosts web pages. A web server is what makes the World Wide Web possible. Each website has one or more web servers. Also, each server can host multiple websites.	Computers with a web browser

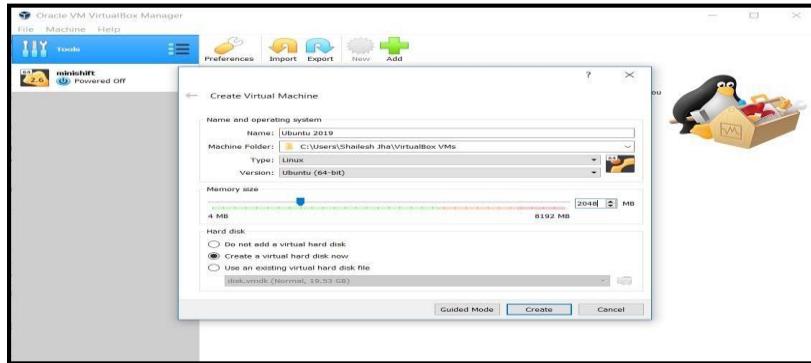
EXPERIMENT NO: 3

INSTALLATION OF LINUX ON VIRTUAL BOX IN WINDOWS 10

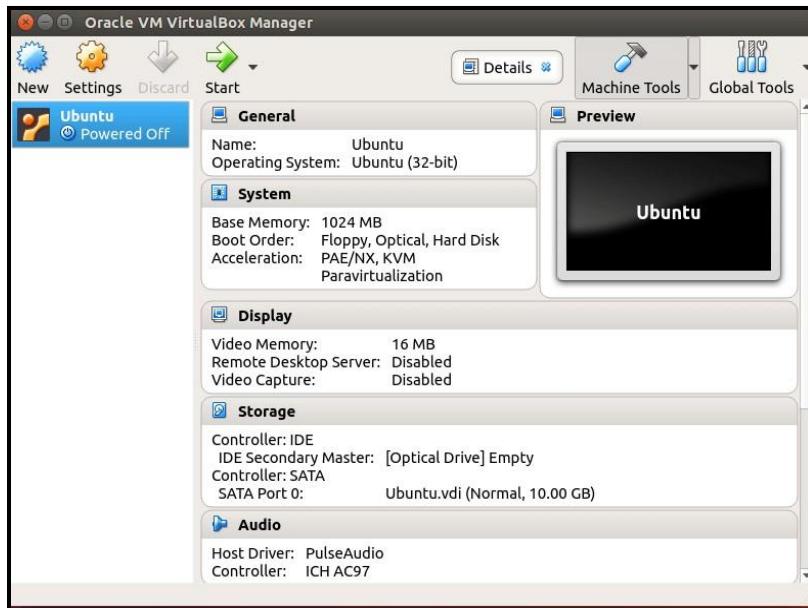
1. Download Virtual Box for Linux Mint from https://www.virtualbox.org/wiki/Linux_Downloads
2. Open the deb file using GDebi Package Installer and install the package.



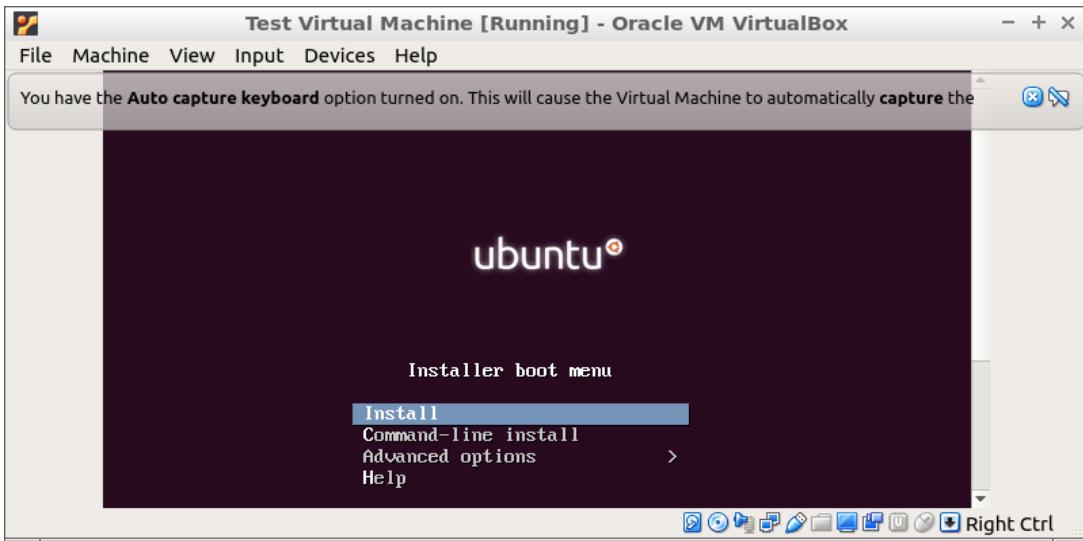
3. Download the ISO file of UBUNTU 20.10 LTS(downloading via BitTorrent is preferred since it is less prone to error and doesn't require verification of ISO file after downloading ascompared todownloading via HTTP)
4. To install lubuntu virtually on the system use Oracle Virtual box(free, open-source cross-platform application for creating, managing and running virtual machines)
5. After installing Oracle virtual box, run the software and click on the button to create a new virtual machine. Then give the necessary information about the UBUNTU in the create virtual machine window.



6. Next, select the memory(RAM) that Oracle VM virtual box should allocate everytime the virtualmachine started. The memory you give the virtual machine will not be available to yourhost OS while the VM is running
7. Next, select the memory(RAM) that Oracle VM virtual box should allocate everytime the virtualmachine started. The memory you give the virtual machine will not be available to yourhost OS while the VM is running
8. Then insert the lubuntu 20.10 LTS ISO file(Storage → Click the empty option → Click CD/DVDicon → Select the ISO file downloaded earlier → Click Ok)
9. Then select the ubuntu Virtual Machine and click on the Start button



10. During ubuntu startup “no bootable medium found-system halted” error messages will occur sometimes if so abort the booting and repeat step 5 & 6
11. After successful booting, when ubuntu desktop interface is displayed, double click on Install



EXPERIMENT NO:4

INSTALLATION OF WINDOWS ON VIRTUAL BOX IN UBUNTU

Step 1: Download Windows 10 ISO

First and foremost, you need to download a Windows 10 ISO. You can download Windows 10 32-bit or 64-bit, depending on your system.

Step 2: Install VirtualBox on Ubuntu

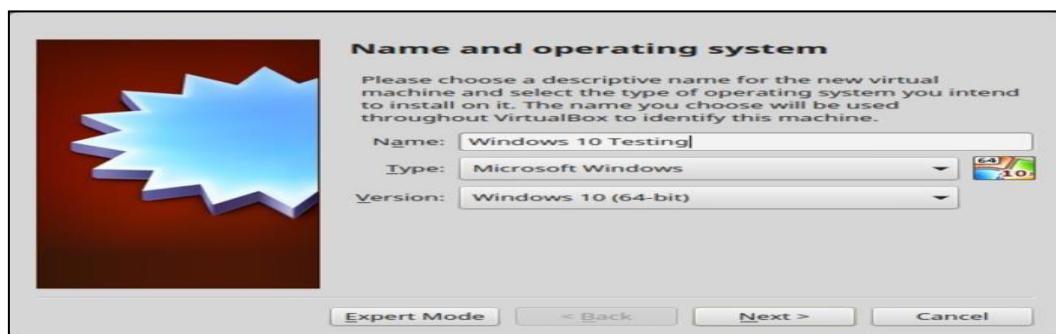
Install VirtualBox on Ubuntu using the command `sudo apt install virtualbox`

Step 3: Install Windows 10 in VirtualBox

- Start VirtualBox. You should see a screen like the one below.

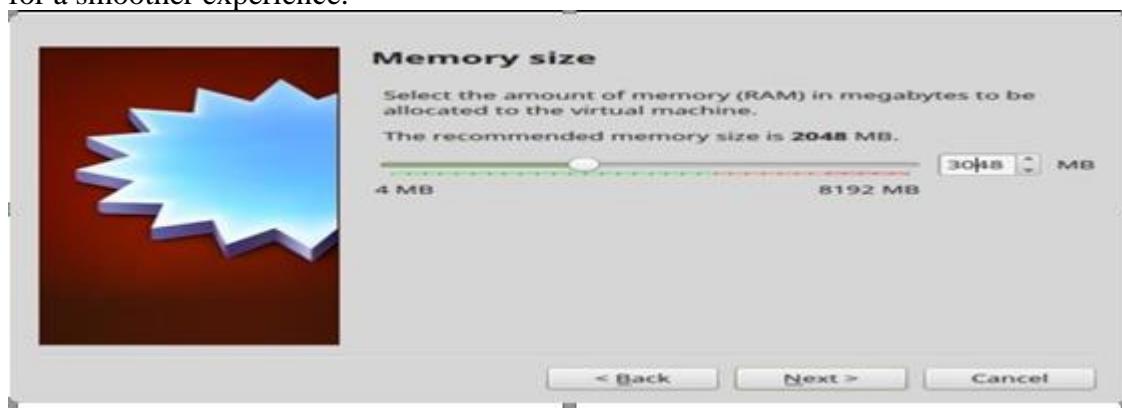


- Click on new and name the VM anything you like. Also select the operating system and version. In this case I've chosen Windows 10 and 64-bit.



- Choose the RAM size. The recommended RAM size for 64-bit Windows 10 is 2 GB,

while for 32-bit Windows 10 it's 1 GB. But I suggest you have a little more than that for a smoother experience.



- Next is the size for the virtual machine. If you're crunched on space choose the recommended size, otherwise make it a little more than the recommended size.



- Select Create a virtual disk now, for the format, go ahead with the VDI format.



- Click on start to install Windows 10

EXPERIMENT NO:5

INSTALLATION OF WINDOWS OPERATING SYSTEM ON A BARE MACHINE

STEP 1 : ACQUIRING REQUIRED OS/OEM

Before installing Windows 10, we must decide which OS version image we should install in our computer. Most users have the option to choose between the official Windows image from Microsoft or the official OEM Image from their Machine manufacturer like. HP, Dell, Asus and so on.

Acquiring Official Windows 10 OS from Microsoft :

1. Go to the Windows 10 official page and select the latest build of the os in dropdown menu and press ‘confirm’.(<https://www.microsoft.com/en-us/softwaredownload/windows10ISO>).
2. A new dropdown menu appears which allows us to choose the preferred language for our OS. Please choose the required language from the same and press ‘confirm’.
3. Next we see a menu to download the 32-bit & 64-bit architecture versions of the OS for download. Please verify your hardware and download the required version.
4. The download takes time as the image file is usually 5-6 GB in size. Once the download is complete, we will have file with extension (.iso).

STEP 2: PREPARING BOOTABLE MEDIUM

To install the OS onto a machine, we need a bootable medium containing the installation files. The two common media are bootable Flash drive & bootable DVD. We must prepare either one of them in order to install the OS.

Preparing Bootable DVD:

In order to create a bootable medium, we must burn the downloaded disk image of the OS (.iso) onto an empty DVD. Use either built-in functionality or 3rd-party software for the same.

Preparing Bootable Pen Drive:

- 1) Download the Rufus software from the official website.
- 2) Open the Rufus software and the below UI opens.
- 3) Insert an empty flash drive with at least 8GB space into your computer and it will appear in the RufusUI. Now, click the 'Select' button and choose the downloaded Windows OS Image(.iso) and click 'start' to initiate the process. Remember the flash drive will be formatted on conversion to bootable medium.
- 4) After the process finishes, the software will notify us the same and our flash drive is now successfully converted into a bootable medium.

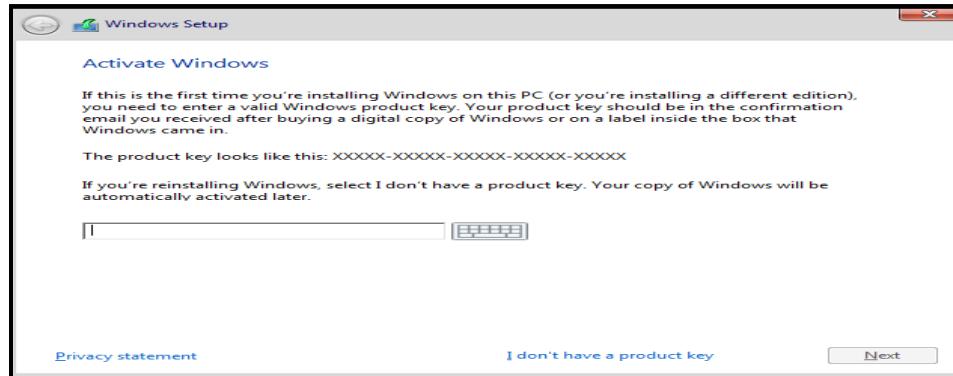
STEP 3 : SYSTEM BOOTING & INSTALLATION

Once the bootable medium is ready, we can use it to boot the machine/PC and install the OS to the hard-disk.

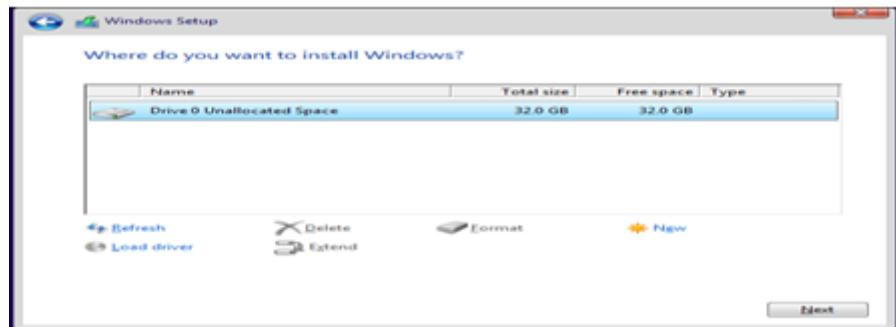
1. Insert the bootable DVD or flash drive into the computer and restart it.
2. Usually the BIOS will automatically boot using the bootable medium, otherwise manually boot it up on the boot menu.
3. Choose the required language and layout settings in the initial installation window and press next and install button.



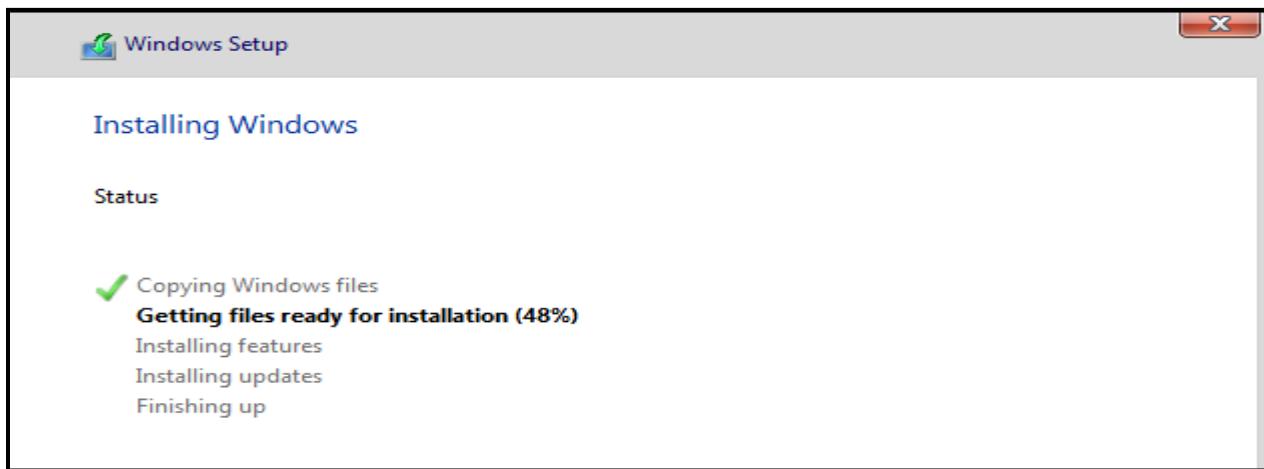
4. Now the installer will prompt you to input a valid License Key. If you do not have one, you can purchase and input it later.
5. Next choose the version of Windows 10 you like to install. Make sure you purchase or have a matching License Key.



6. Accept the software license terms by ticking the checkbox and click next.
7. If you want to upgrade an existing version of Windows, select upgrade option, else select CustomInstall for a fresh installation.
8. Next, inside the disk partition screen, you can create disk partitions and select the drive to install windows which will become the C:\ Drive.



1. On clicking next, the installer starts to install the Windows OS to the Hard Disk, please wait as it will take a few minutes to complete the same.



2. On complete installation, the PC will restart and move to a initial screen to configure your region, keyboard layout, time zone settings. Once they are setup, you will be introduced to the Windows 10 Desktop Screen.
3. Now, make sure you update the windows system immediately download and install updates to install necessary drivers and software patches to the system. In case of some proprietary hardware, the user mayhave to install the drivers manually.
4. Monitor your system performance on Task manager, if everything is working normally, your installation is complete and your system is ready for use.

EXPERIMENT NO:6

INSTALLING LINUX OVER WINDOWS OPERATING SYSTEM

REQUIREMENTS:

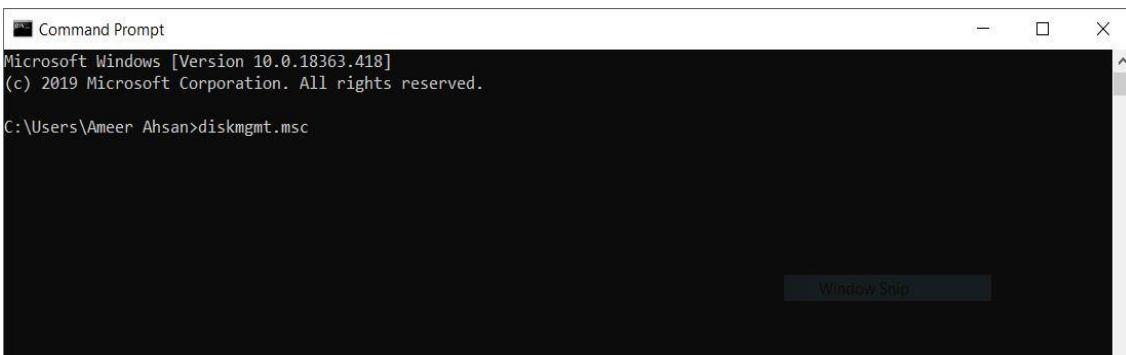
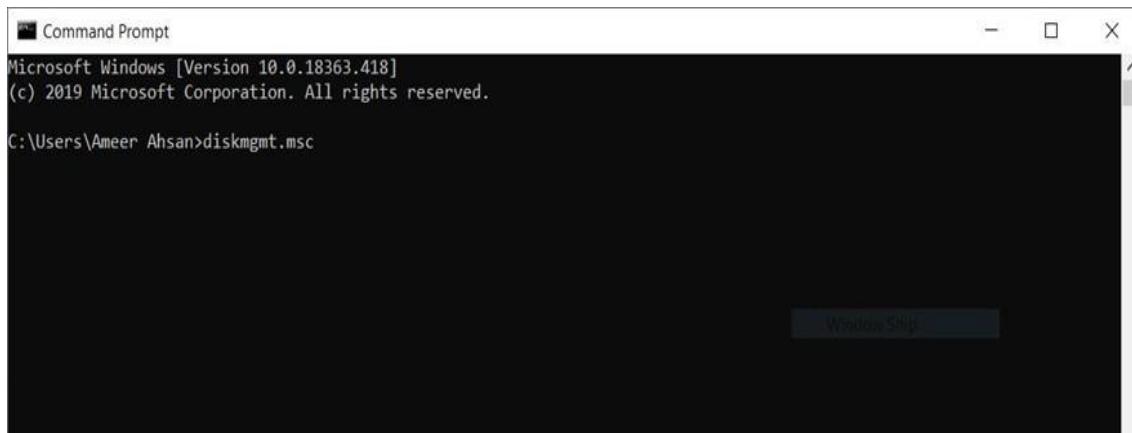
- ❖ PC with at least 2GB RAM
- ❖ 8GB Flash Drive
- ❖ Rufus
- ❖ Linux ISO file

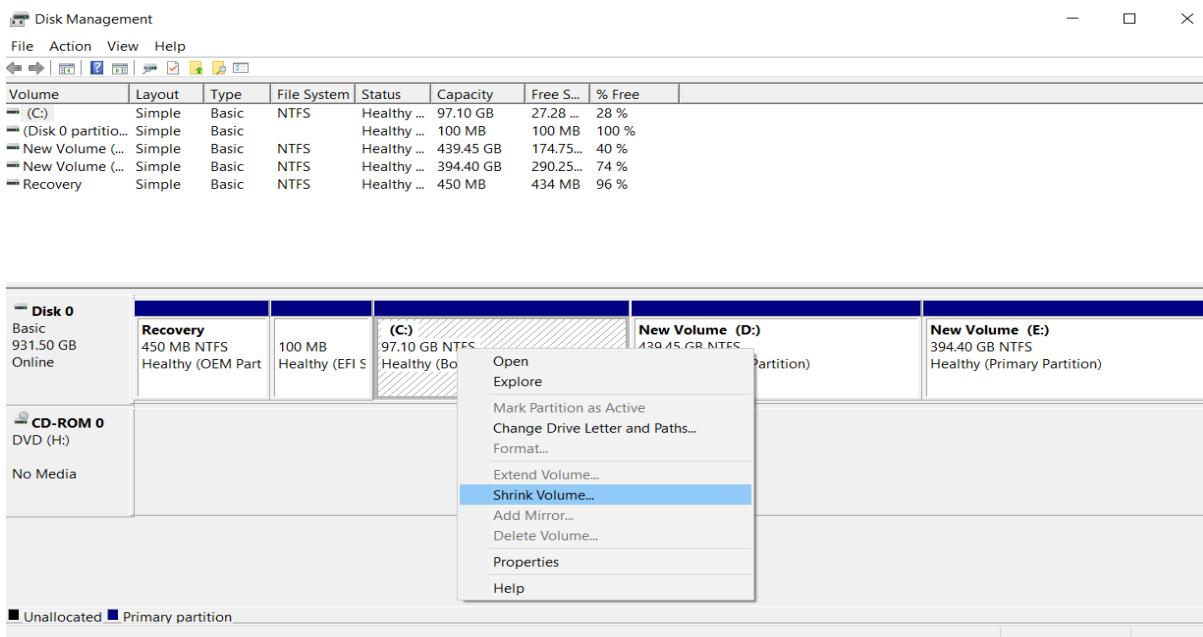
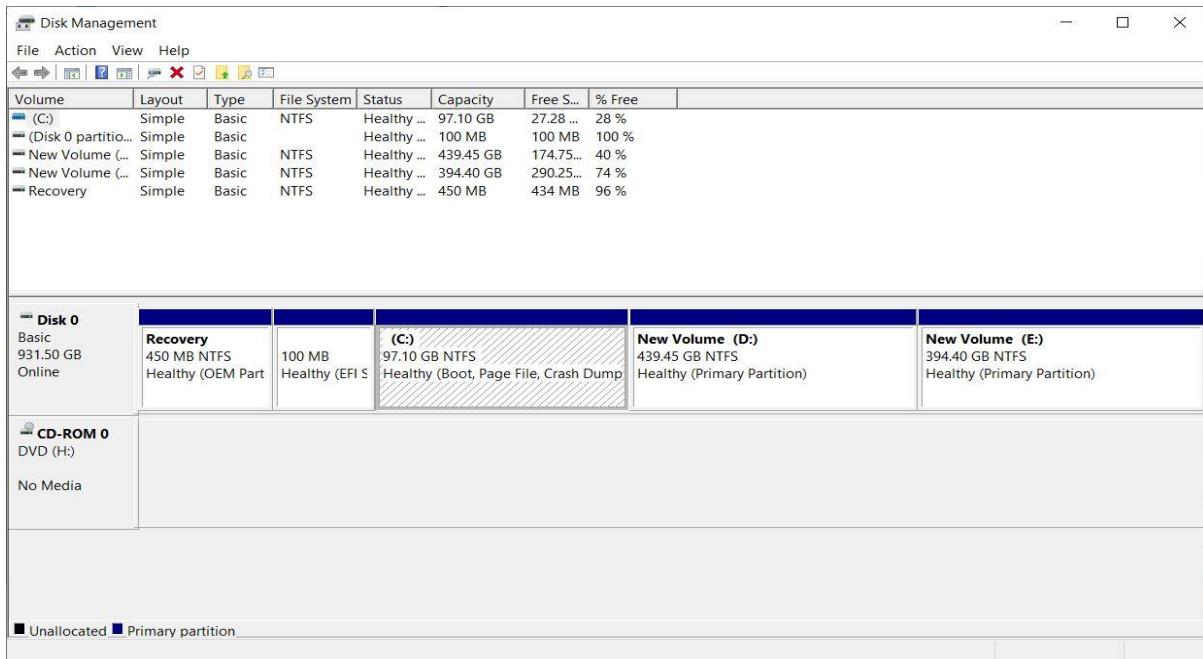
STEPS INVOLVED:

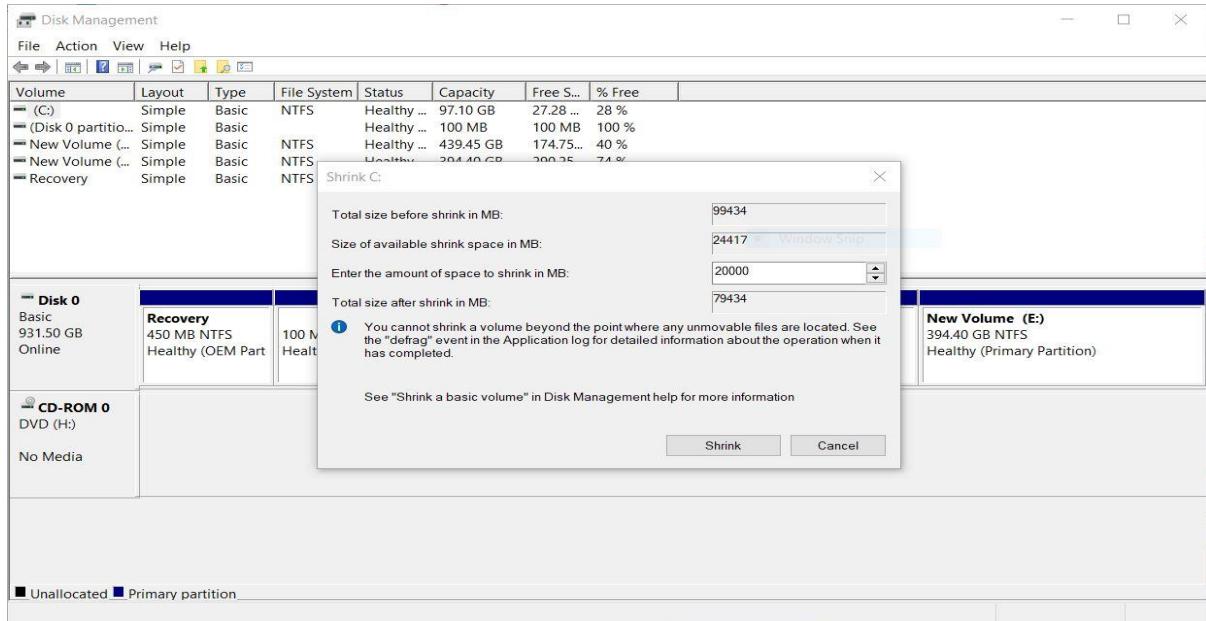
- ❖ Partition of hard drive in windows10
- ❖ Make a Linux Bootable USB
- ❖ Installation of Linux ISO from USB

STEP.1: Partition of Hard Drive in Windows

1. Open cmd
2. Type the command “diskmgmt.msc” and hit Enter.







Make a Linux Bootable USB

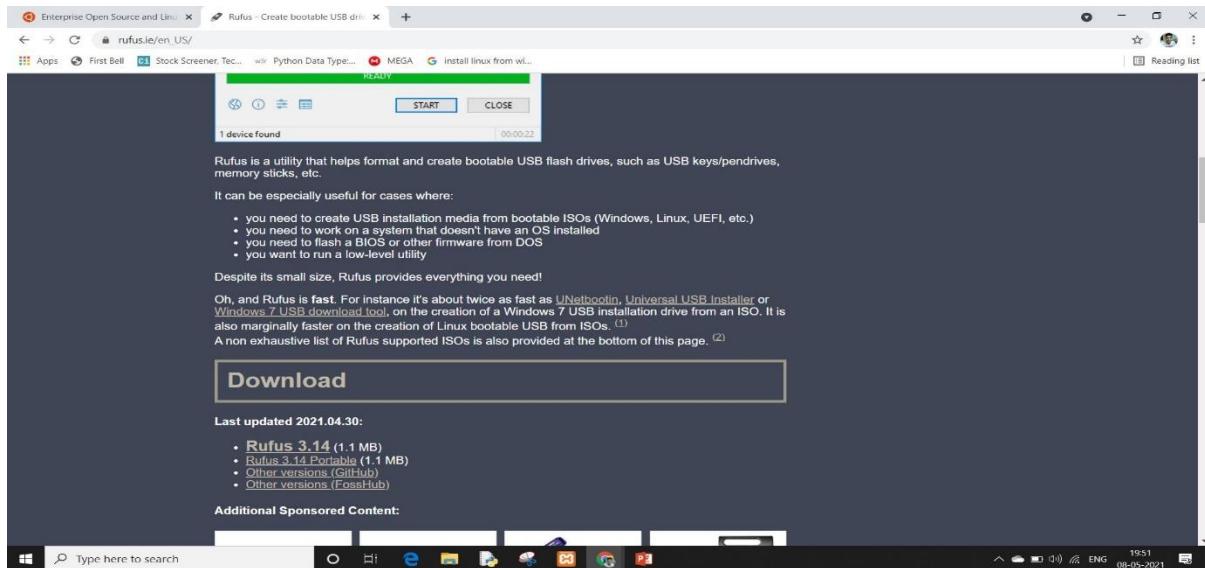
1. Download a Linux distro in ISO format.

<https://ubuntu.com/>

Insert the USB drive into your computer.

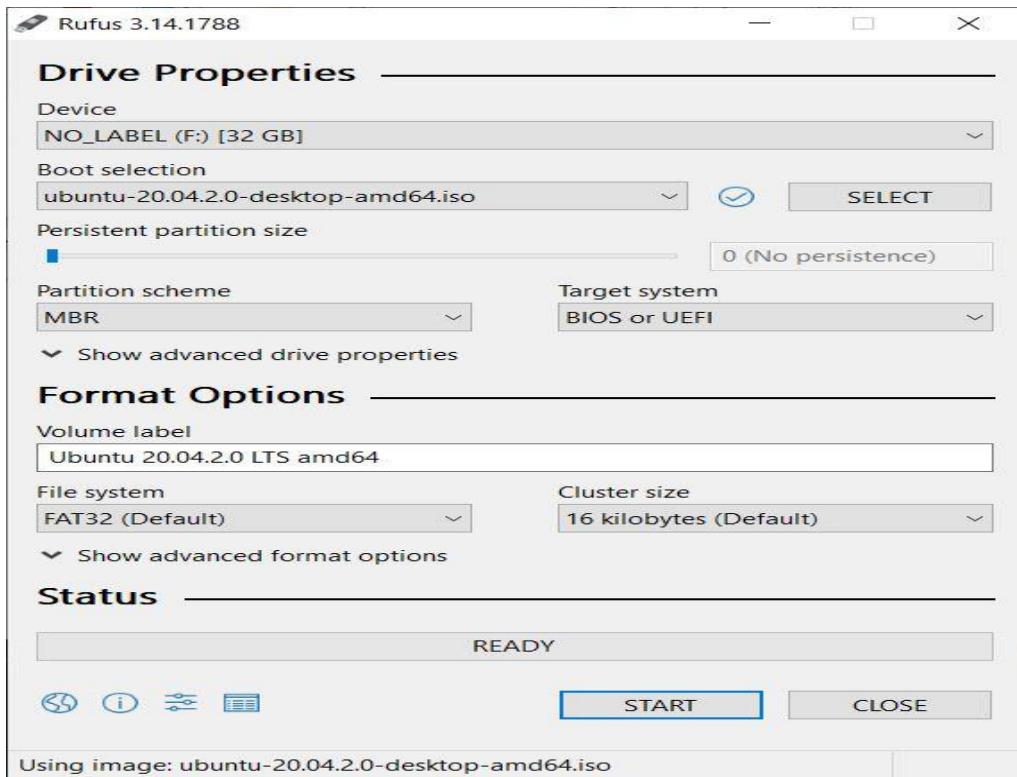
3. Download Rufus

https://rufus.ie/en_US/



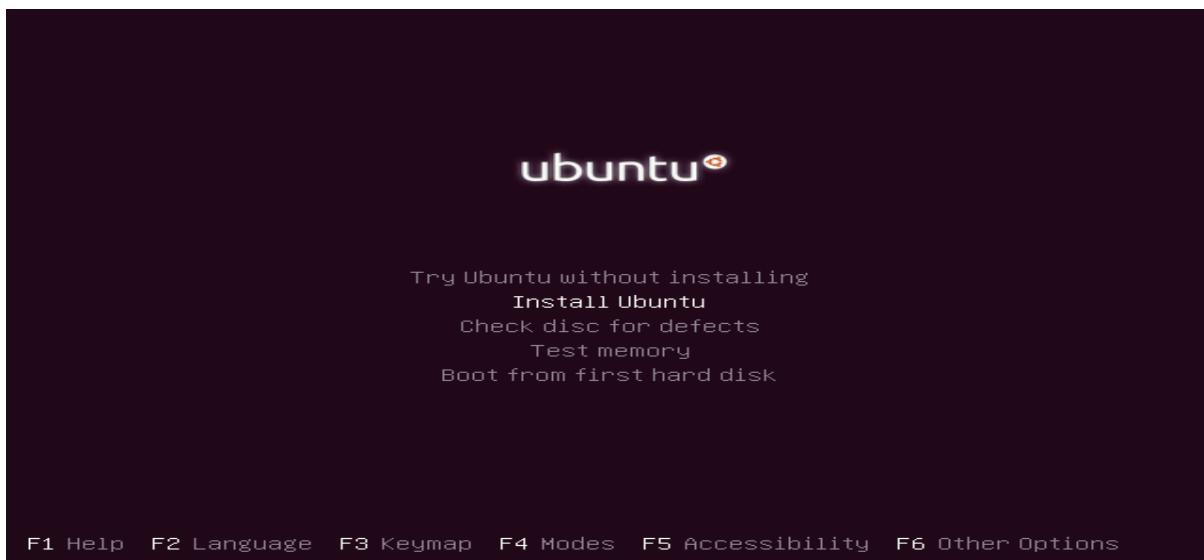
4. Open Rufus and select your USB drive from the Device list

5. Under Boot Selection, click the Select button and choose the ISO file you downloaded

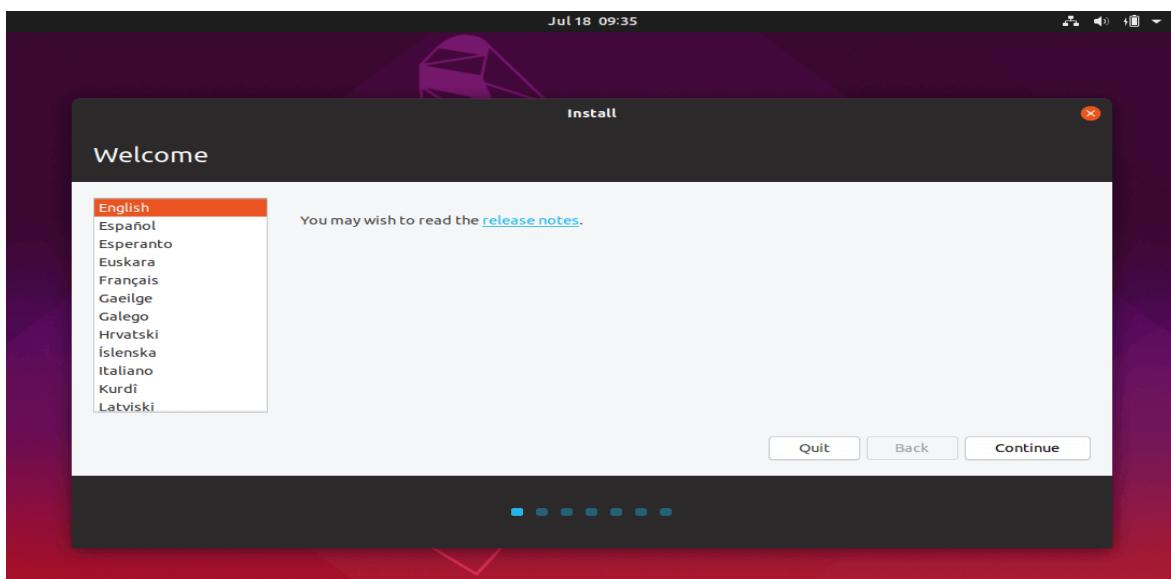


Install Linux from USB

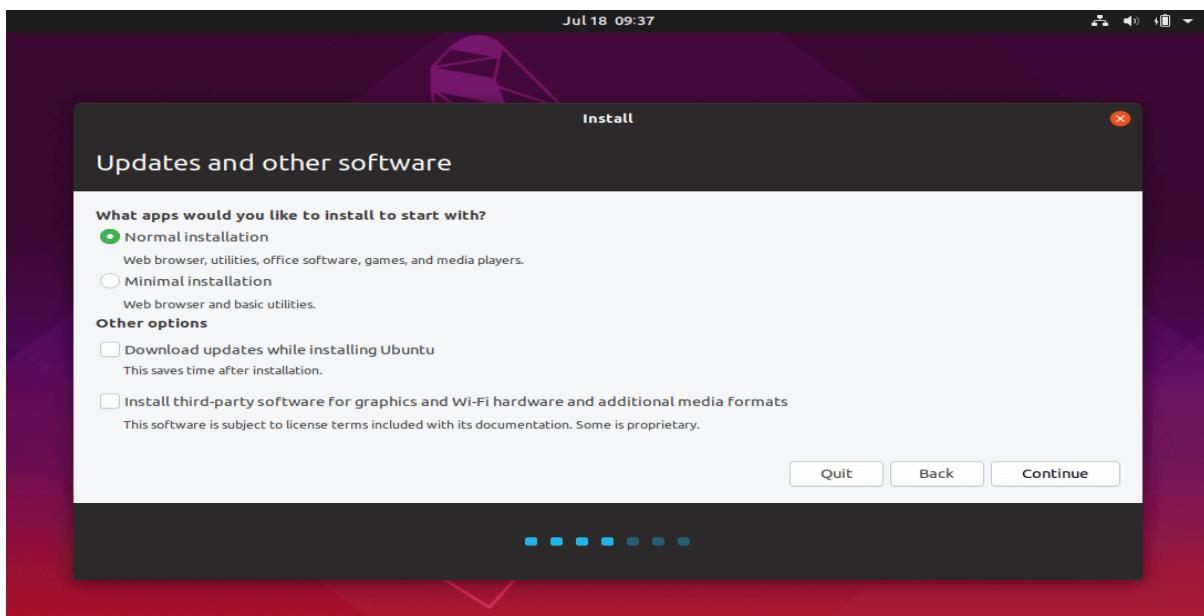
1. Place the USB stick, reboot the machine and instruct the **UEFI** to boot-up from the USB by pressing a special function key (usually **F12**, **F10** or **F2** depending on the vendor specifications).



2. Choose the language you wish to perform the installation and click on the **Continue**

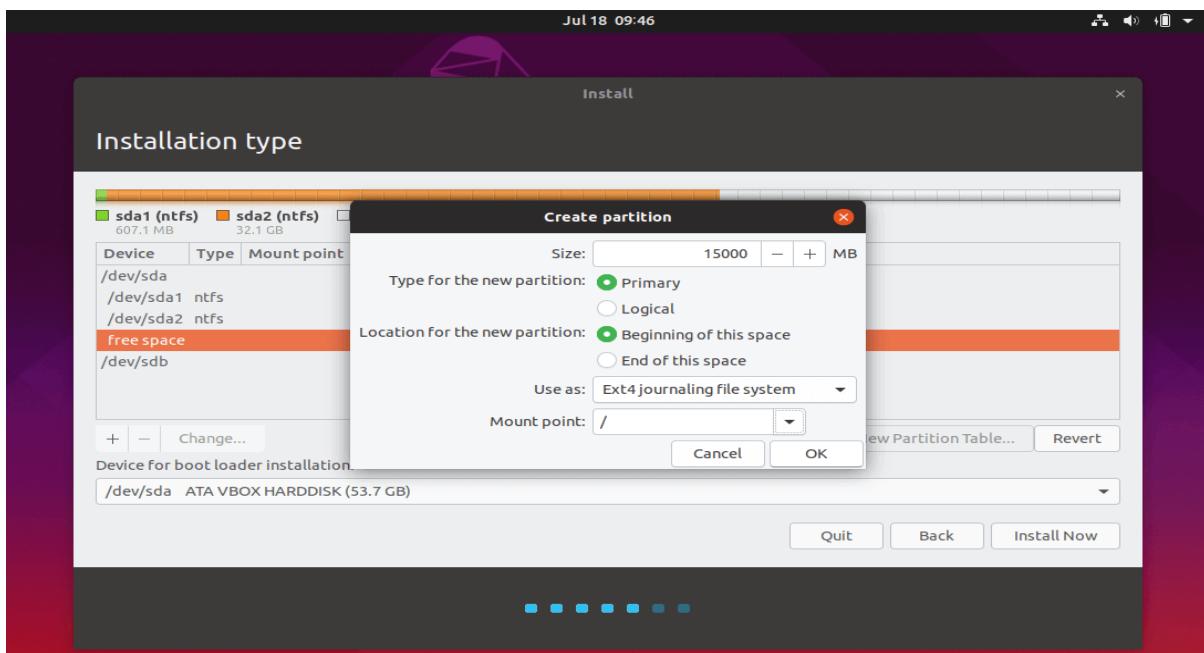


3. choose the first option “Normal Installation” and hit on the **Continue** button

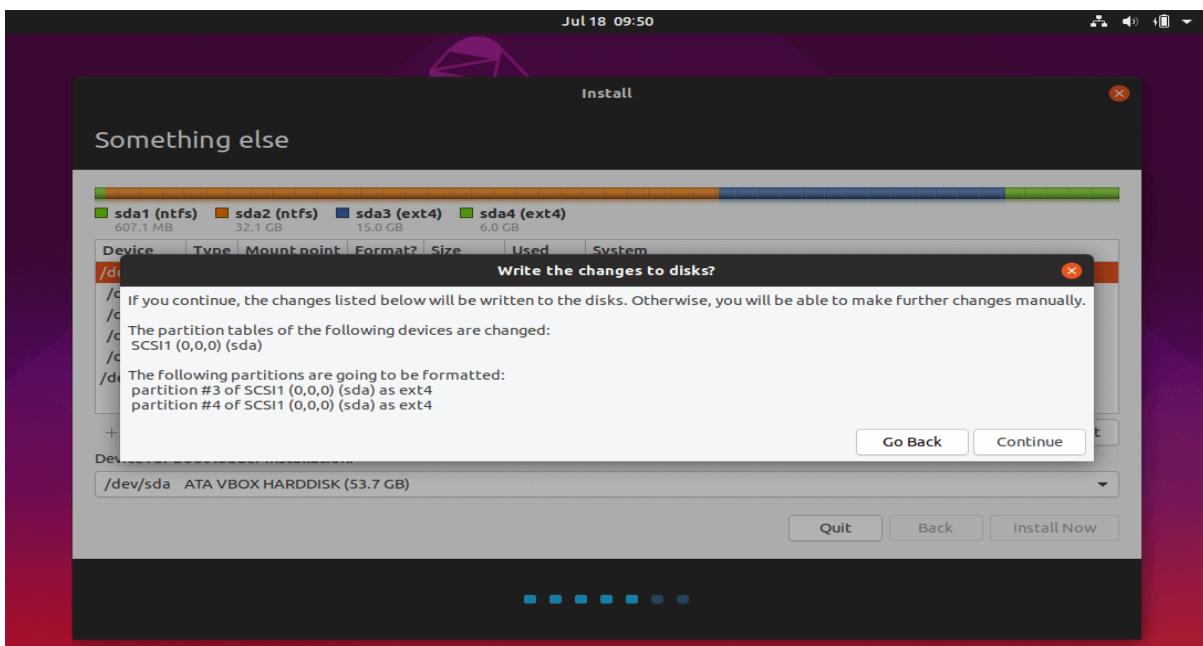


4. check the **Something else** option and hit on the **Continue** button

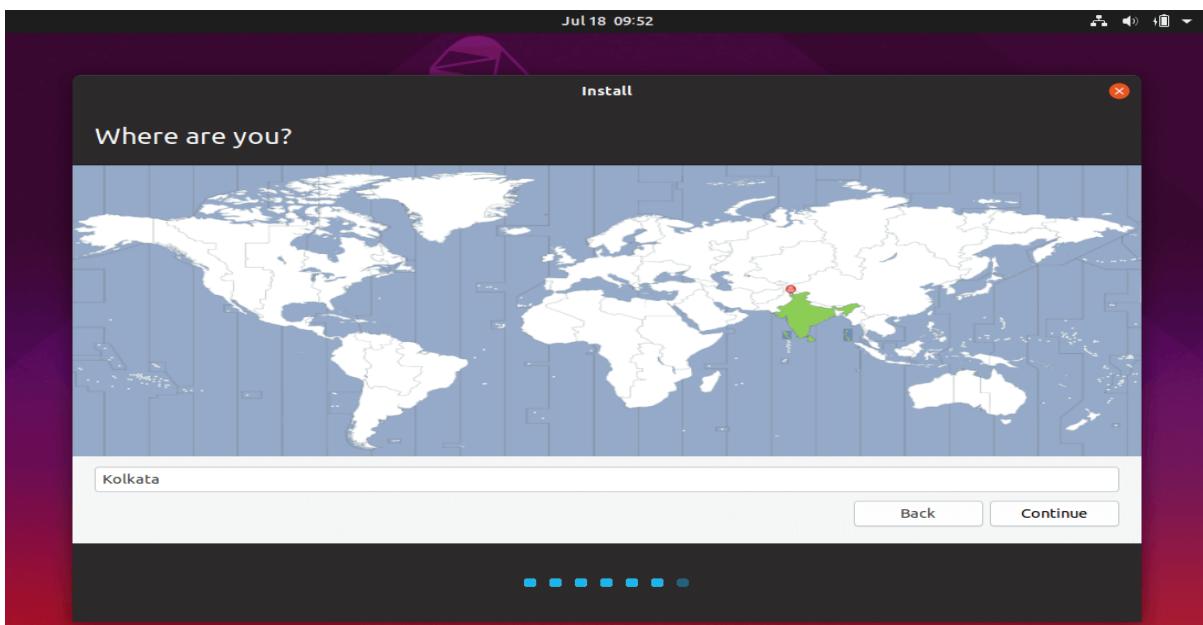
5. we'll create our custom partition layout for **Ubuntu**



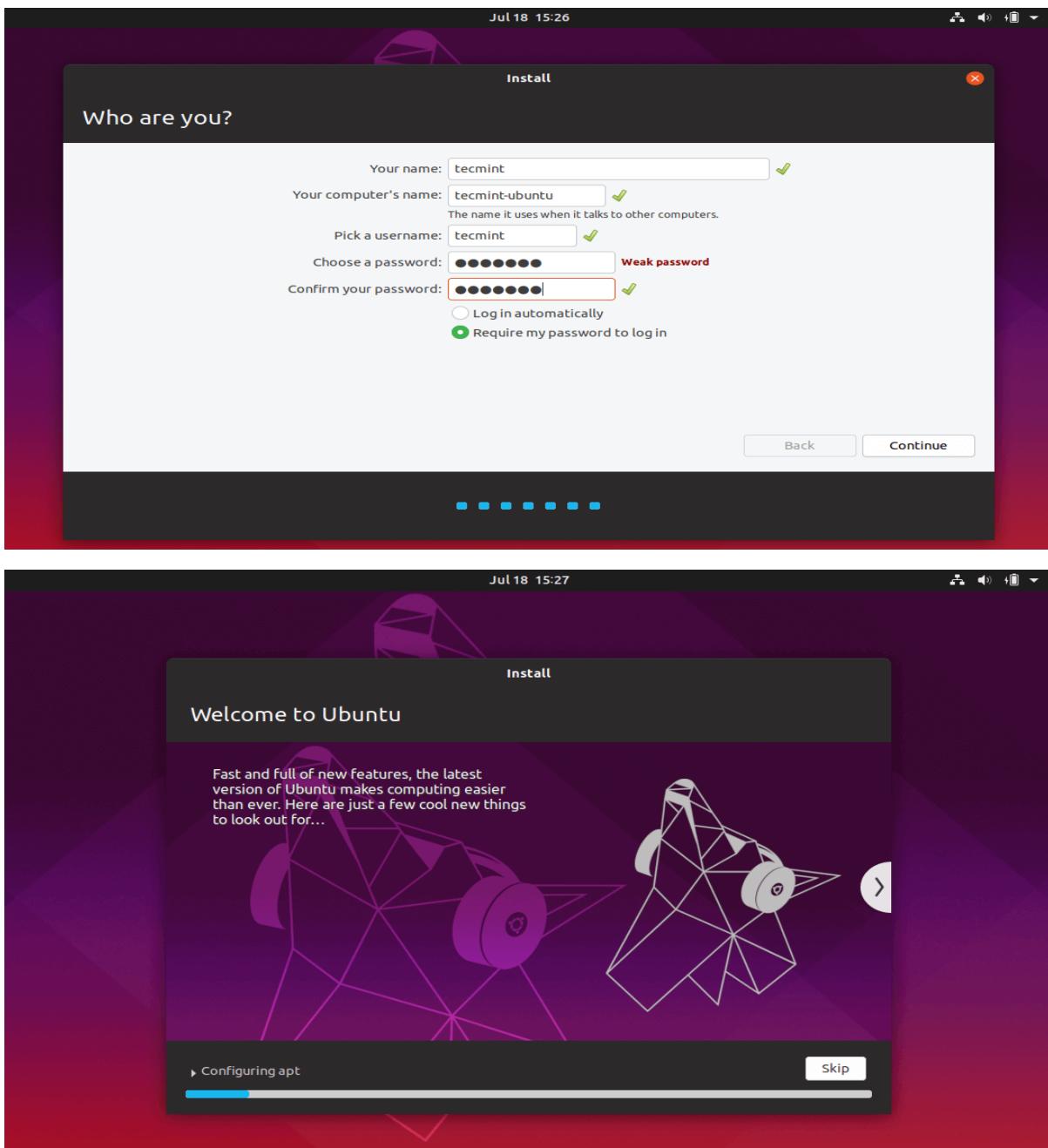
6.hit the **Install Now** button in order to apply changes to disk and start the installation process.



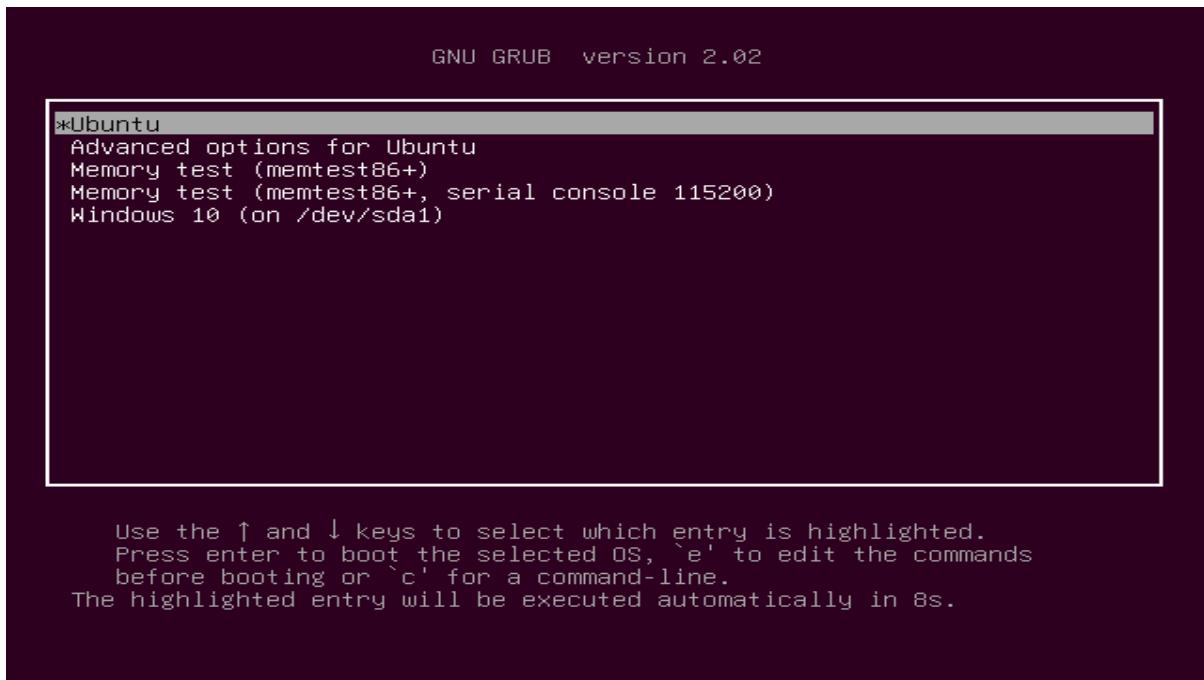
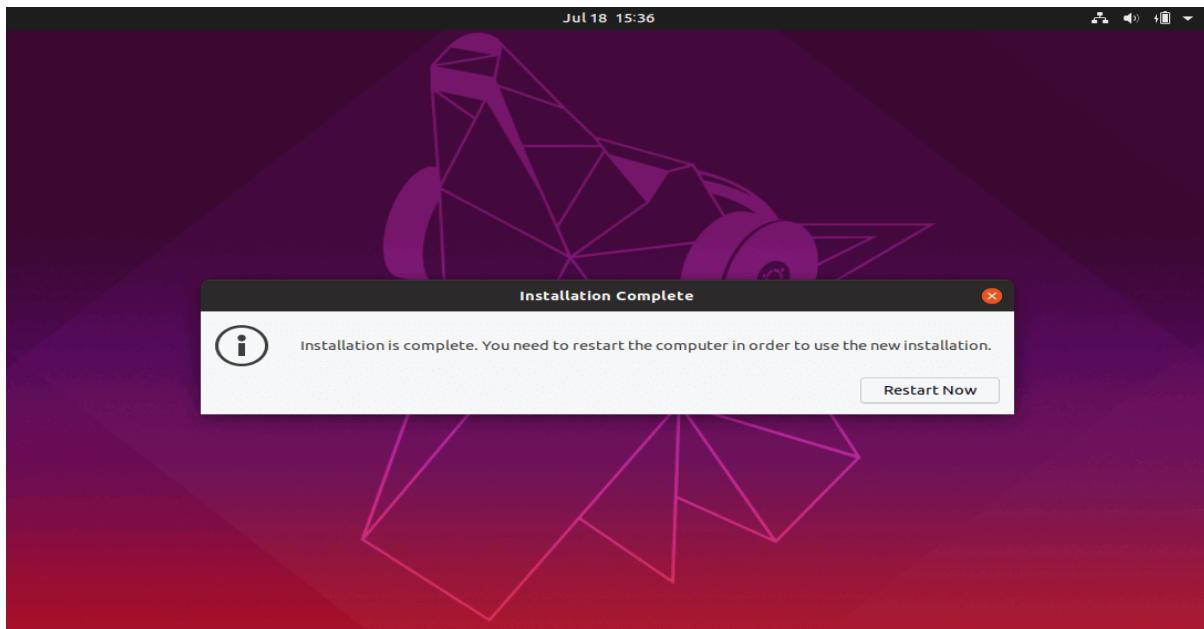
7.set your machine physical location by selecting a city nearby from the map. When done hit **Continue** to move ahead.



8. Pick up a username and password for your administrative **sudo** account, enter a descriptive name for your computer and hit **Continue** to finalize the installation.



9. After the installation process reaches its end hit on the **Restart Now** button in order to complete the installation.



EXPERIMENT NO: 7

Installation of server system

1 – Planning the installation

When planning the installation of the server operating system it's necessary to ensure the right hardware is in place. Windows Server 2012 is a 64-bit operating system, so it cannot be installed on a server running 32-bit processors.

With this latest version, there are only two versions of the operating system, Standard and Datacenter. We are going to go with the Standard version (there is no feature difference between the two versions, only a difference in support for hardware and guest virtual machines).

Also, this will be a brand new install, so we don't have to worry about backing up data and applications (although the former is always a good idea in case of future failures)

2 – Initial set-up

Assuming that an installation disc has been provided, insert the disc into the computer and switch on the server. When prompted, press any key to boot from the disc. The files will start loading and the next screen will show the Language dialogue box.

Select the correct language, currency and time options as well as the keyboard you wish to use. Click on next. Then another dialogue box will show an Install Now button, click on that.

We also have to decide what type of install to make. Server Core has a minimal installation footprint and is great for tasks where it's not necessary to use a graphical user interface (GUI) – the onscreen 'click and drag' which most of us use every day – but for the purpose of this guide, we will install the OS with a full GUI. So click on the tick (check) box to accept the licence terms and click "Next".

On the next screen you are asked to choose what type of install you want. This is a fresh install so we will select "Custom: Install Windows only (advanced)"

3 – Installing

Then you will be asked where you want to install Windows. In this case, we are installing the OS on the primary partition. (You can also use the Drive options link on the dialogue box to configure any drives attached to the server if this is necessary.)

Then Windows will begin installing itself to your server. How long this takes will depend on your server's specifications. Once it is completed you will be asked to provide an administrator password for the new installation. Type in a password and click "Next".

4 – Managing the server and changing computer name

Once this is done, press Ctrl-Alt-Delete at the same time to get the login screen. Type in the administrator password and press enter.

It is important to note here that as Server 2012 takes its design cues from Windows 8, most tasks can also be carried out via the Start Screen, which can be accessed by pressing the Windows button on the keyboard.

Open Server Manager by clicking on its tile. Windows Server then changes from Modern UI to desktop mode, and the Server Manager window will appear on this desktop. In this window click on "Configure this local server". This allows you to set up the various features you will require for your network.

Most servers are given a generic name, but this can be changed to something that will match any naming convention you have in mind (especially true when this server is the first of many). Renaming is the first option on the Server Manager window.

To change the server name, click on the existing one in the Server Manager app. When the System Properties comes up, click on the "Change" button. Type in a new name in the box and click on "OK". You will have to restart after doing this.

5 – Configure Server IP address

If you don't use DHCP – the protocol for configuring devices connected to a network – for this server, you have also to set a static IP address, default gateway and domain name system. To do this, this click on "Ethernet" and the Network Connection windows will open.

Double click on the adaptor you wish to change IP addresses to bring up the information page. Then click on "Properties" and double-click on "Internet Protocol version 4". You can then type in all the IP address information for the server.

6 – Managing remotely with Remote Desktop

If you put your server in a fairly inaccessible place, administrating it at the actual machine itself may not always be possible. Setting up Remote Desktop means that you can control the server from another computer inside or outside the office.

In Server Manager, click the Disabled link next to the Remote Desktop entry. In the Server Manager Remote page, choose the radio button next to "Allow Remote Connections to this Computer".

To add extra users so that they can access the server, click on the "Add" button.

7 – Keeping your server updated

Updates on Windows come thick and fast, so it's a good idea to make sure that your server stays up to date with as little input from yourself as possible.

From the Server Manager app, click on "Not Configured" next to Windows Update, following which the Windows Update window will appear. Then click on the "Turn On Automatic Updates button. The server will automatically begin downloading and installing updates to be applied to your machine.

You can also set this to "Download updates but let me choose whether to install them". That gives the administrator the chance to install updates during planned downtime.

8 – Activate Windows Server 2012

Lastly, in Server Manager you can activate the server. Click on the option to do so and then enter the product key that came with the software. Once this is done, your server is configured and ready to go.

EXPERIMENT NO: 8

STUDY ON COMMAND LINE TEXT EDITORS

Text editors are Programs used to create and edit plain text files. Main purpose: to create a file to be used by another program.

for example:

- Hypertext Markup Language (HTML) for a Web browser

- Source code that a compiler

can process There are Two

types of text editors:

- Terminal based

editors

Examples: vi, pic

o, nano, emacs

- GUI editors

Examples: gedit, kwrite

Emacs Editor

Emacs is a very popular editor on Unix until recently. Emacs has so many available features like a calculator, calendar, email client. Commands in emacs are either control characters (hold down the <Ctrl>key while typing another character). Syntax: emacs filename

Pico Editor

The UNIX Pico editor is a full screen editor which is very

easy to use. Syntax: pico filename

Nano Editors

The nano is an improved open source of pico available for

GNU/Linux. Syntax: nano filename

Vim Editor

The vi command-line text editor is included with most versions of UNIX and Linux. Vim is

an improved version of vi. The vi commands are now linked to the vim commands.

There are three modes in vi editor:

There are three modes in vi editor:

- Command mode: Use key combinations as commands instead of typing text.
- Insert mode: Typed text is displayed on screen.
- Extend mode: Used for more advanced commands, such as saving files, exiting vim, or searching and replacing text.

EXPERIMENT NO: 9

Basic Linux Commands

Here is a list of basic Linux commands students are expected to know. You may take this set as a starting point and explore each of them.

1. `pwd`

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 (/).

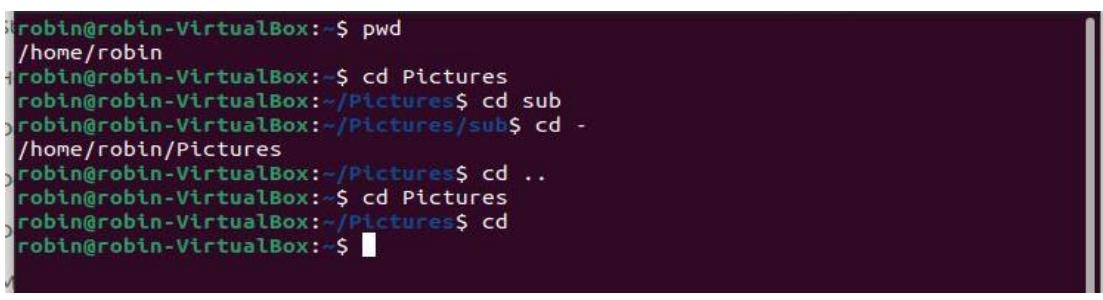
An example of an absolute path is /home/username.

2. `cd`

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

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



```
robin@robin-VirtualBox:~$ pwd
/home/robin
robin@robin-VirtualBox:~$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ cd sub
robin@robin-VirtualBox:~/Pictures/sub$ cd -
/home/robin/Pictures
robin@robin-VirtualBox:~/Pictures$ cd ..
robin@robin-VirtualBox:~$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ cd
robin@robin-VirtualBox:~$
```

3. ls

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.

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.
- **ls -t** lists files sorted in the order of “last modified”
- **-r** option will reverse the natural sorting order. Usually used in combination with other switches such as **ls -tr**. This will reverse the time-wise listing

```
robin@robin-VirtualBox:~$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ ls
sub  text.c  vote.c
robin@robin-VirtualBox:~/Pictures$ ls -R
.:
sub  text.c  vote.c

./sub:
integer.c
robin@robin-VirtualBox:~/Pictures$ ls -a
.  ..  sub  text.c  vote.c
robin@robin-VirtualBox:~/Pictures$ ls -al
total 20
drwxr-xr-x  3 robin robin 4096 Jun 11 13:03 .
drwxr-xr-x 18 robin robin 4096 Jun  9 18:35 ..
drwxrwxr-x  2 robin robin 4096 Jun 11 12:40 sub
-rw-rw-r--  1 robin robin   263 Nov 24  2020 text.c
-rw-rw-r--  1 robin robin   245 Nov 24  2020 vote.c
robin@robin-VirtualBox:~/Pictures$ ls -t
sub  vote.c  text.c
robin@robin-VirtualBox:~/Pictures$ ls -r
vote.c  text.c  sub
robin@robin-VirtualBox:~/Pictures$
```

4. cat

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**

```
robin@robin-VirtualBox:~/Pictures$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ cat vote.c
#include <stdio.h>
int main()
{
    int n1;
    printf("Enter your age: ");
    scanf("%d", &n1);
    if (n1 >= 0)
        printf("You are eligible for voting");

    else
        printf("You are not eligible for voting");

    return 0;
}robin@robin-VirtualBox:~/Pictures$ cat >abc.c
hello World
welcome to Linux
```

```
robin@robin-VirtualBox:~/Pictures$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ cat vote.c abc.c >text.c
robin@robin-VirtualBox:~/Pictures$ cat text|tr a-z A-Z >output.txt
cat: text: No such file or directory
robin@robin-VirtualBox:~/Pictures$ cat text.c|tr a-z A-Z >output.txt
robin@robin-VirtualBox:~/Pictures$
```

5. cp

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.

- **cp -i** will ask for user's consent in case of a potential file overwrite.
- **cp -p** will preserve source files' mode, ownership and timestamp.
- **cp -r** will copy directories recursively.

- cp -u copies files only if the destination file is not existing or the source file is newer than the destination file.

6.mv

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 is **mv oldname.ext newname.ext**

```
robin@robin-VirtualBox:~/Pictures$ cp text.c /home/robin/Documents
robin@robin-VirtualBox:~/Pictures$ cp -i text.c /home/robin/Documents
cp: overwrite '/home/robin/Documents/text.c'?
robin@robin-VirtualBox:~/Pictures$ cp -r text.c /home/robin/Documents
robin@robin-VirtualBox:~/Pictures$ cp -u text.c /home/robin/Documents
robin@robin-VirtualBox:~/Pictures$ mv vote.c /home/robin/Documents
robin@robin-VirtualBox:~/Pictures$
```

7. mkdir

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.

```
robin@robin-VirtualBox:~/Pictures$ mkdir empfolder
robin@robin-VirtualBox:~/Pictures$
```

8. rmdir

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

```
robin@robin-VirtualBox:~/Pictures$ rmdir empfolder
robin@robin-VirtualBox:~/Pictures$ █
```

9. rm

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.

```
robin@robin-VirtualBox:~/Pictures$ mkdir Sample
robin@robin-VirtualBox:~/Pictures$ mkdir Sample/Sub
robin@robin-VirtualBox:~/Pictures$ mkdir -p Sample/2021/Sub
robin@robin-VirtualBox:~/Pictures$ mkdir sample1
robin@robin-VirtualBox:~/Pictures$ rmdir sample1
robin@robin-VirtualBox:~/Pictures$ rm -r newfile
robin@robin-VirtualBox:~/Pictures$ █
```

10. touch

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 anHTML file entitled **Web** under the **Documents** directory.



11. locate

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.

```
robin@robin-VirtualBox:~/Pictures$ locate hello.txt
/home/robin/Pictures/hello.txt
robin@robin-VirtualBox:~/Pictures$
```

12. find

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**

```
robin@robin-VirtualBox:~/Pictures$ cd ..
robin@robin-VirtualBox:~$ find /home -name computer.txt
/home/robin/Music/computer.txt
robin@robin-VirtualBox:~$
```

13. grep

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. You should refer to some grep tutorial

Useful for command line use as well. Usually output of a previous command is piped into the grep command. For example ls -l | grep “kernel”

```
robin@robin-VirtualBox:~/Music$ cat computer.txt
Hello, welcome to the world
robin@robin-VirtualBox:~/Music$ grep world computer.txt
Hello, welcome to the world
robin@robin-VirtualBox:~/Music$
```

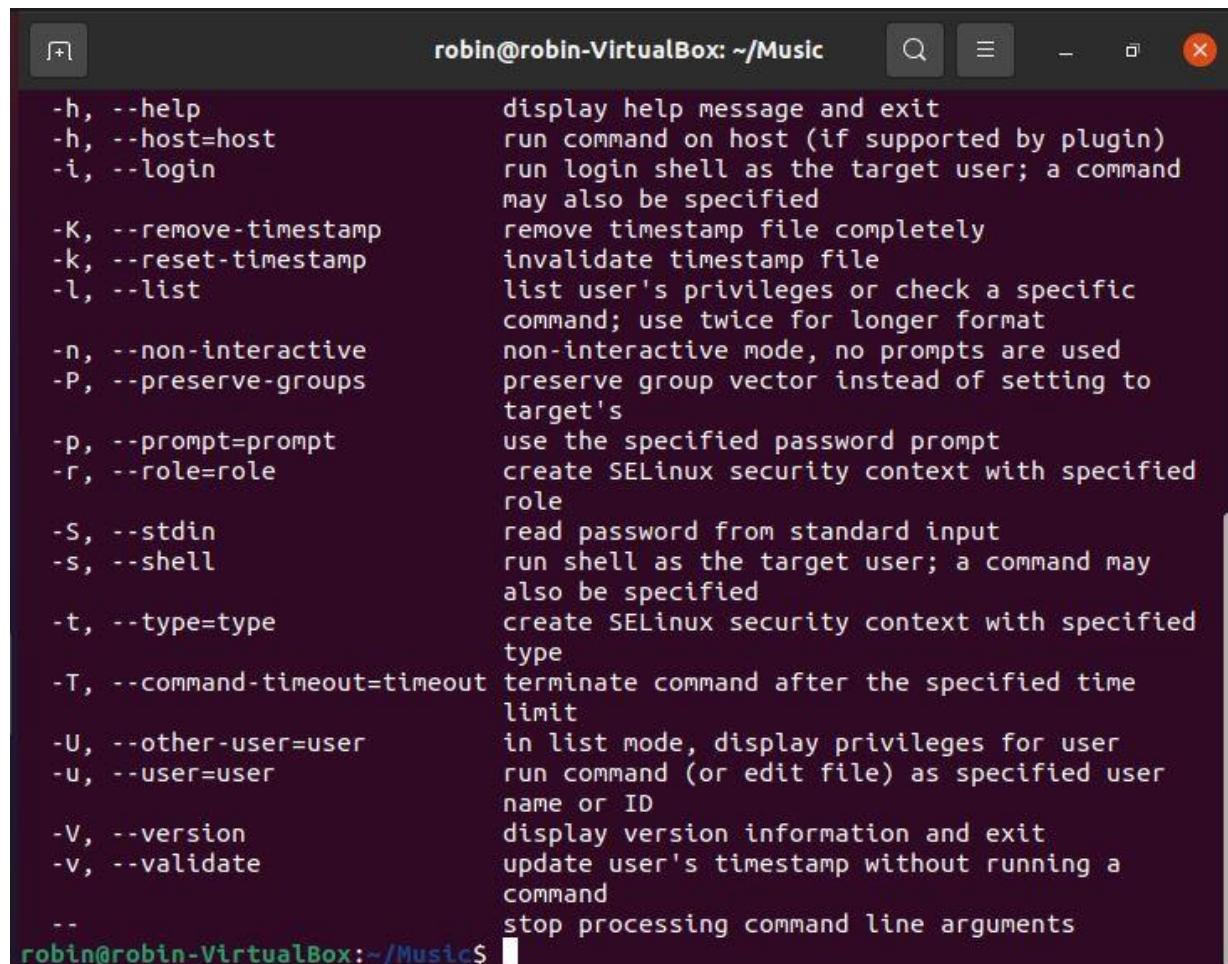
14. sudo

Short for “**SuperUser Do**”, this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command.

```
robin@robin-VirtualBox:~/Music$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
      [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
      prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
      prompt] [-T timeout] [-u user] file ...

Options:
  -A, --askpass           use a helper program for password prompting
  -b, --background        run command in the background
  -B, --bell              ring bell when prompting
  -C, --close-from=num   close all file descriptors >= num
  -E, --preserve-env     preserve user environment when running command
  --preserve-env=list    preserve specific environment variables
  -e, --edit              edit files instead of running a command
  -g, --group=group       run command as the specified group name or ID
  -H, --set-home          set HOME variable to target user's home dir
  -h, --help               display help message and exit
  -h, --host=host         run command on host (if supported by plugin)
  -i, --login             run login shell as the target user; a command
                         may also be specified
  -K, --remove-timestamp remove timestamp file completely
  -k, --reset-timestamp  invalidate timestamp file
  -l, --list              list user's privileges or check a specific
```

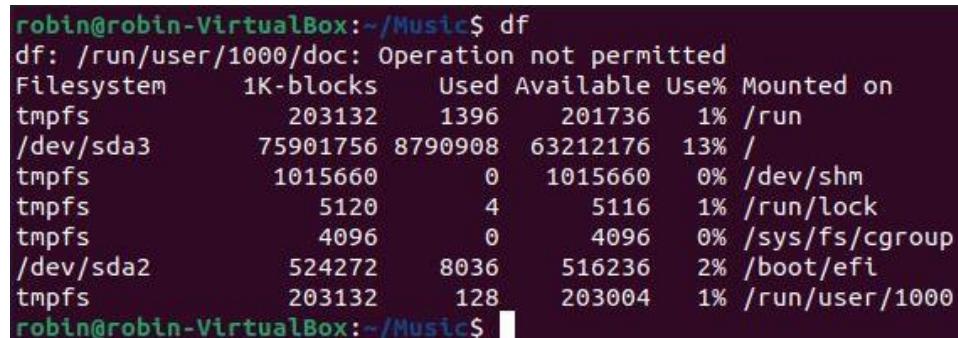


The terminal window shows the man page for the 'runuser' command. The title bar reads 'robin@robin-VirtualBox: ~/Music'. The page lists various options and their descriptions:

- h, --help**: display help message and exit
- h, --host=host**: run command on host (if supported by plugin)
- i, --login**: run login shell as the target user; a command may also be specified
- K, --remove-timestamp**: remove timestamp file completely
- k, --reset-timestamp**: invalidate timestamp file
- l, --list**: list user's privileges or check a specific command; use twice for longer format
- n, --non-interactive**: non-interactive mode, no prompts are used
- P, --preserve-groups**: preserve group vector instead of setting to target's
- p, --prompt=prompt**: use the specified password prompt
- r, --role=role**: create SELinux security context with specified role
- S, --stdin**: read password from standard input
- s, --shell**: run shell as the target user; a command may also be specified
- t, --type=type**: create SELinux security context with specified type
- T, --command-timeout=timeout**: terminate command after the specified time limit
- U, --other-user=user**: in list mode, display privileges for user
- u, --user=user**: run command (or edit file) as specified user name or ID
- V, --version**: display version information and exit
- v, --validate**: update user's timestamp without running a command
- : stop processing command line arguments

15. df

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**.



```
robin@robin-VirtualBox:~/Music$ df
df: /run/user/1000/doc: Operation not permitted
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs            203132     1396   201736   1% /run
/dev/sda3        75901756  8790908  63212176  13% /
tmpfs           1015660       0   1015660   0% /dev/shm
tmpfs             5120        4     5116   1% /run/lock
tmpfs              4096        0     4096   0% /sys/fs/cgroup
/dev/sda2        524272     8036   516236   2% /boot/efi
tmpfs            203132     128   203004   1% /run/user/1000
robin@robin-VirtualBox:~/Music$
```

```
robin@robin-VirtualBox:~$ df -m
df: /run/user/1000/doc: Operation not permitted
Filesystem      1M-blocks  Used Available Use% Mounted on
tmpfs            199       2      198   1% /run
/dev/sda3        74123    8585    61731  13% /
tmpfs            992       0      992   0% /dev/shm
tmpfs              5       1       5   1% /run/lock
tmpfs              4       0       4   0% /sys/fs/cgroup
/dev/sda2         512       8      505   2% /boot/efi
tmpfs            199       1      199   1% /run/user/1000
robin@robin-VirtualBox:~$
```

16. du

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.

```
robin@robin-VirtualBox:~$ du
8      ./Music
4      ./Desktop/empdict
8      ./Desktop
4      ./ssh
4      ./Templates
4      ./gnupg/private-keys-v1.d
16     ./gnupg
4      ./Public
4      ./Downloads
4      ./config/update-notifier
8      ./config/ibus/bus
12     ./config/ibus
4      ./config/goa-1.0
12     ./config/dconf
8      ./config/yelp
4      ./config/nautilus
8      ./config/gtk-3.0
4      ./config/enchant
84     ./config/pulse
8      ./config/gedit
8      ./config/evolution/sources
12     ./config/evolution
4      ./config/gnome-session/saved-session
8      ./config/gnome-session
184    ./config
24     ./snap/snap-store/common/.cache/gio-modules
68     ./snap/snap-store/common/.cache/immodules
64     ./snap/snap-store/common/.cache/fontconfig
```

```
robin@robin-VirtualBox:~$ du -h
8.0K  ./Music
4.0K  ./Desktop/empdict
8.0K  ./Desktop
4.0K  ./ssh
4.0K  ./Templates
4.0K  ./gnupg/private-keys-v1.d
16K   ./gnupg
4.0K  ./Public
4.0K  ./Downloads
4.0K  ./config/update-notifier
8.0K  ./config/ibus/bus
12K   ./config/ibus
4.0K  ./config/goa-1.0
12K   ./config/dconf
8.0K  ./config/yelp
4.0K  ./config/nautilus
8.0K  ./config/gtk-3.0
4.0K  ./config/enchant
84K   ./config/pulse
8.0K  ./config/gedit
8.0K  ./config/evolution/sources
12K   ./config/evolution
4.0K  ./config/gnome-session/saved-session
8.0K  ./config/gnome-session
184K  ./config
24K   ./snap/snap-store/common/.cache/gio-modules
68K   ./snap/snap-store/common/.cache/immodules
64K   ./snap/snap-store/common/.cache/fontconfig
```

17. head

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. (Read the manual)

```
robin@robin-VirtualBox:~/Music$ cat computer.txt
Hello, welcome to the world
Enjoy yourself
robin@robin-VirtualBox:~/Music$ head -n 1 computer.txt
Hello, welcome to the world
robin@robin-VirtualBox:~/Music$
```

18. tail

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.

```
robin@robin-VirtualBox:~/Music$ cat computer.txt
Hello, welcome to the world
Enjoy yourself
here we have
tiger
Lion
cat
dog
human
goat
deer
birds
save planet
robin@robin-VirtualBox:~/Music$ tail computer.txt
here we have
tiger
Lion
cat
dog
human
goat
deer
birds
save planet
robin@robin-VirtualBox:~/Music$
```

19. diff

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

```
robin@robin-VirtualBox:~/Music$ cat computer.txt
tiger
Lion
cat
dog
human

robin@robin-VirtualBox:~/Music$ cat file.txt
dog
cat
tiger
Zebra
robin@robin-VirtualBox:~/Music$ diff computer.txt file.txt
0a1,2
> dog
> cat
2,6c4
< Lion
< cat
< dog
< human
<
---
> Zebra
robin@robin-VirtualBox:~/Music$
```

20. tar

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. Read some tutorial on net.

```
robin@robin-VirtualBox:~/Music$ tar -cvf test.tar test
test/
test/file.txt
test/computer.txt
robin@robin-VirtualBox:~/Music$ tar -xvf test.tar test
test/
test/file.txt
test/computer.txt
robin@robin-VirtualBox:~/Music$
```

21. chmod

chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. Read about permissions and how to manipulate them .

```
robin@robin-VirtualBox:~/Music$ ls -l
total 20
-rw-rw-r-- 1 robin robin 21 Sep 8 16:46 file.txt
drwxrwxr-x 2 robin robin 4096 Sep 8 17:19 test
-rw-rw-r-- 1 robin robin 10240 Sep 8 17:16 test.tar
robin@robin-VirtualBox:~/Music$ ls -l file.txt
-rw-rw-r-- 1 robin robin 21 Sep 8 16:46 file.txt
robin@robin-VirtualBox:~/Music$ chmod u=rw,og=r file.txt
robin@robin-VirtualBox:~/Music$ ls -l file.txt
-rw-r--r-- 1 robin robin 21 Sep 8 16:46 file.txt
robin@robin-VirtualBox:~/Music$
```

22. chown

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.

```
robin@robin-VirtualBox:~/Music$ ls -l file.txt
-rwxr-xr-x 1 robin robin 21 Sep  8 16:46 file.txt
robin@robin-VirtualBox:~/Music$ chown root file.txt
chown: changing ownership of 'file.txt': Operation not permitted
robin@robin-VirtualBox:~/Music$
```

23. ps

Ps command will display all current processes along with their process ids (PID) . Read manuals for various options.

```
robin@robin-VirtualBox:~/Music$ ps
  PID TTY      TIME CMD
 2499 pts/0    00:00:00 bash
 4672 pts/0    00:00:00 ps
robin@robin-VirtualBox:~/Music$
```

```
robin@robin-VirtualBox:~/Music$ ps -ux
USER      PID %CPU %MEM   VSZ   RSS TTY      STAT START  TIME COMMAND
robin     763  0.0  0.5 20352 10820 ?
robin     764  0.0  0.1 170428 3628 ?
robin     770  0.0  0.9 2204952 19576 ?
robin     772  0.0  1.2 520844 25012 ?
robin     775  0.0  0.3 248852 6800 ?
robin     779  0.0  0.3 173524 6640 tty2
robin     781  0.3  3.7 849720 76912 tty2
robin     784  0.0  0.3 10036 6212 ?
robin     792  0.0  0.3 248388 7824 ?
robin     807  0.0  0.3 378344 6532 ?
robin     818  0.0  0.4 323688 9480 ?
robin     824  0.0  0.3 244320 6348 ?
robin     828  0.0  0.3 322960 7640 ?
robin     834  0.0  0.3 246592 6896 ?
robin     838  0.0  0.3 244516 6236 ?
robin     842  0.0  1.8 629280 36796 ?
robin     859  0.0  0.4 323440 8836 ?
robin     883  0.0  0.6 198164 13972 tty2
robin     955  0.0  0.0 6036 444 ?
robin     982  0.0  0.3 305400 6760 ?
robin     987  0.0  0.2 8280 4164 ?
robin    1010  0.0  0.2 99564 4540 ?
robin    1017  0.0  0.7 698916 16088 ?
robin    1036  0.9 17.2 4254388 350204 ?
robin    1071  0.0  0.5 322512 11156 ?
robin    1075  0.0  0.3 171360 7124 ?
robin    1076  0.0  1.4 283880 29904 ?
```

24. Kill

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.

You can issue kill -9 PID

25. ping

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.

```
robin@robin-VirtualBox:~$ ping google.com
PING google.com (142.250.67.174) 56(84) bytes of data.
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=1 ttl=111 time=74.0 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=2 ttl=111 time=80.0 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=3 ttl=111 time=88.7 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=4 ttl=111 time=76.7 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=5 ttl=111 time=100 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=6 ttl=111 time=80.8 ms
64 bytes from bom12s07-in-f14.1e100.net (142.250.67.174): icmp_seq=7 ttl=111 time=87.1 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6014ms
rtt min/avg/max/mdev = 74.013/83.935/100.266/8.236 ms
robin@robin-VirtualBox:~$
```

26. wget

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.

```
robin@robin-VirtualBox:~$ wget https://www.oracle.com/in/index.html
--2021-09-08 17:56:08-- https://www.oracle.com/in/index.html
Resolving www.oracle.com (www.oracle.com)... 104.85.110.192, 2405:200:1630:4b8:a15, 2405:200:1630:487::a15
Connecting to www.oracle.com (www.oracle.com)|104.85.110.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘index.html.1’

index.html.1 [=>] 42.33K --.-KB/s in 0.06s

2021-09-08 17:56:09 (668 KB/s) - ‘index.html.1’ saved [43345]

robin@robin-VirtualBox:~$
```

27. uname

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.

```
robin@robin-VirtualBox:~$ uname
Linux
robin@robin-VirtualBox:~$
```

28. top

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.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1036	robin	20	0	4254388	351464	125680	S	2.7	17.3	1:52.41	gnome-+
4853	robin	20	0	364988	16560	14212	R	1.0	0.8	0:00.03	tracke+
524	syslog	20	0	221176	4716	3680	S	0.3	0.2	0:00.38	rsyslo+
781	robin	20	0	849720	76976	43916	S	0.3	3.8	0:42.11	Xorg
784	robin	20	0	10036	6212	3752	S	0.3	0.3	0:02.13	dbus-d+
1183	robin	20	0	1171408	30144	20800	D	0.3	1.5	0:00.67	gsd-me+
4684	robin	20	0	824320	49216	37796	S	0.3	2.4	0:01.47	gnome-+
1	root	20	0	168684	11976	8776	S	0.0	0.6	0:04.49	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthrea+
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_pa+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_per+
10	root	20	0	0	0	0	S	0.0	0.0	0:00.11	ksoftti+
11	root	20	0	0	0	0	I	0.0	0.0	0:01.75	rcu_sc+
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.13	migrat+
13	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_i+
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_i+
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.39	migrat+
18	root	20	0	0	0	0	S	0.0	0.0	0:00.26	ksoftti+

29. history

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 s you've entered before.

```
robin@robin-VirtualBox: $ history
1 tracerouter
2 traceroute
3 sudo apt install traceroute
4 clear
5 traceroute google.com
6 clear
7 traceroute -m 5 google.com
8 clear
9 route
10 clear
11 sudo nano /etc/netplan/01-network-manager-all.yaml
12 ip config all
13 ipconfig all
14 ipconfig -a
15 ip config -a
16 ip a
17 sudo nano /etc/netplan/01-network-manager-all.yaml
18 sudo netplan apply
19 ip a
20 sudo ufw status
21 clear
22 sudo ufw status
23 sudo ufw enable
24 sudo ufw status
25 sudo ufw disable
26 sudo ufw status
27 clear
28 sudo ufw disable
```

30. man

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.

Use the command: man man to start learning about man utility.

```
robin@robin-VirtualBox:~$ man
What manual page do you want?
For example, try 'man man'.
robin@robin-VirtualBox:~$ man man
```

31. echo

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

```
robin@robin-VirtualBox:~$ echo Welcome to our World
Welcome to our World
robin@robin-VirtualBox:~$
```

32. zip, unzip

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. (This program should be installed , some distributions may not have them. You can also look at gzip and bzip commands)

```
robin@robin-VirtualBox:~$ zip
Copyright (c) 1990-2008 Info-ZIP - Type 'zip "-L"' for software license.
Zip 3.0 (July 5th 2008). Usage:
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
The default action is to add or replace zipfile entries from list, which
can include the special name - to compress standard input.
If zipfile and list are omitted, zip compresses stdin to stdout.
-f  freshen: only changed files -u  update: only changed or new files
-d  delete entries in zipfile   -m  move into zipfile (delete OS files)
-r  recurse into directories   -j  junk (don't record) directory names
-0  store only                -l  convert LF to CR LF (-ll CR LF to LF)
-1  compress faster           -9  compress better
-q  quiet operation            -v  verbose operation/print version info
-c  add one-line comments     -z  add zipfile comment
-@  read names from stdin    -o  make zipfile as old as latest entry
-x  exclude the following names -i  include only the following names
-F  fix zipfile (-FF try harder) -D  do not add directory entries
-A  adjust self-extracting exe -J  junk zipfile prefix (unzipsfx)
-T  test zipfile integrity    -X  eXclude extra file attributes
-y  store symbolic links as the link instead of the referenced file
-e  encrypt                   -n  don't compress these suffixes
-h2 show more help
```

robin@robin-VirtualBox:~\$ █

```
robin@robin-VirtualBox:~$ unzip
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.

Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
Default action is to extract files in list, except those in xlist, to exdir;
file[.zip] may be a wildcard. -Z => ZipInfo mode ("unzip -Z" for usage).

-p  extract files to pipe, no messages      -l  list files (short format)
-f  freshen existing files, create none     -t  test compressed archive data
-u  update files, create if necessary       -z  display archive comment only
-v  list verbosely/show version info        -T  timestamp archive to latest
-x  exclude files that follow (in xlist)    -d  extract files into exdir

modifiers:
-n  never overwrite existing files          -q  quiet mode (-qq => quieter)
-o  overwrite files WITHOUT prompting       -a  auto-convert any text files
-j  junk paths (do not make directories)    -aa treat ALL files as text
-U  use escapes for all non-ASCII Unicode   -UU ignore any Unicode fields
-C  match filenames case-insensitively     -L  make (some) names lowercase
-X  restore UID/GID info                   -V  retain VMS version numbers
-K  keep setuid/setgid/tacky permissions    -M  pipe through "more" pager
-O CHARSET specify a character encoding for DOS, Windows and OS/2 archives
-I CHARSET specify a character encoding for UNIX and other archives

See "unzip -hh" or unzip.txt for more help. Examples:
unzip data1 -x joe  => extract all files except joe from zipfile data1.zip
unzip -p foo | more  => send contents of foo.zip via pipe into program more
unzip -fo foo ReadMe => quietly replace existing ReadMe if archive file newer
robin@robin-VirtualBox:~$
```

33. hostname

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.

```
robin@robin-VirtualBox:~$ hostname
robin-VirtualBox
robin@robin-VirtualBox:~$ hostname -I
10.0.2.15
robin@robin-VirtualBox:~$
```

34. useradd, userdel

(This is available only to system admins) 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.

```
robin@robin-VirtualBox:~$ useradd
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
      --badnames          do not check for bad names
      -b, --base-dir BASE_DIR    base directory for the home directory of the
                                 new account
      --btrfs-subvolume-home   use BTRFS subvolume for home directory
      -c, --comment COMMENT    GECOS field of the new account
      -d, --home-dir HOME_DIR   home directory of the new account
      -D, --defaults           print or change default useradd configuration
      -e, --expiredate EXPIRE_DATE expiration date of the new account
      -f, --inactive INACTIVE   password inactivity period of the new account
      -g, --gid GROUP          name or ID of the primary group of the new
                                 account
      -G, --groups GROUPS      list of supplementary groups of the new
                                 account
      -h, --help               display this help message and exit
      -k, --skel SKEL_DIR      use this alternative skeleton directory
      -K, --key KEY=VALUE      override /etc/login.defs defaults
      -l, --no-log-init        do not add the user to the lastlog and
                                 faillog databases
      -m, --create-home        create the user's home directory
      -M, --no-create-home     do not create the user's home directory
      -N, --no-user-group      do not create a group with the same name as
                                 the user
      -o, --non-unique         allow to create users with duplicate
```

```
robin@robin-VirtualBox:~$ userdel
Usage: userdel [options] LOGIN

Options:
  -f, --force                force removal of files,
                               even if not owned by user
  -h, --help                  display this help message and exit
  -r, --remove                remove home directory and mail spool
                               directory to chroot into
  -R, --root CHROOT_DIR      prefix directory where are located the /etc/* f
  -P, --prefix PREFIX_DIR    iles
  --extrausers                Use the extra users database
  -Z, --selinux-user          remove any SELinux user mapping for the user

robin@robin-VirtualBox:~$
```

EXPERIMENT NO:10

FAMILIARISATION WITH VI EDITOR

- The visual editor is present in Linux.
- The vi editor comes with every version of Linux or Unix.
- Using vi is similar to using other editors in which you can see your file on the screen .
- The vi editor is the most popular editor in linux. The current version is really "vim", but to invoke it simply type "vi".
- Before vi, the primary editor used was the line editor - User was able to see/edit only one line of the text at a time .
- The vi editor is not a text formatter (like MS Word, Word Perfect, etc.) - you cannot set margins - center headings
- It is believed that that vi was originally written by Bill Joy in 1976.
- He co-founded Sun Microsystems in 1982 and served as chief scientist until 2003.
 - The vi editor is:
 - A very powerful
 - It is hard to learn, specially for windows users
 - Move from point to point in the file, and make changes.
 - Available on all UNIX systems
 - Case Sensitive
- To start vi

Type vi at the shell prompt

After pressing enter the command prompt disappears and you see tilde(~) characters on all the lines

These tilde(~) characters indicate that the line is blank



The screenshot shows a terminal window with a dark background. At the top, the title bar displays "robin@robin-VirtualBox: ~". Below the title bar, the command "robin@robin-VirtualBox:~\$ vi test" is visible. The main area of the terminal is a dark purple color, indicating that the file "test" is currently empty. There are no tilde (~) characters present in the terminal window.

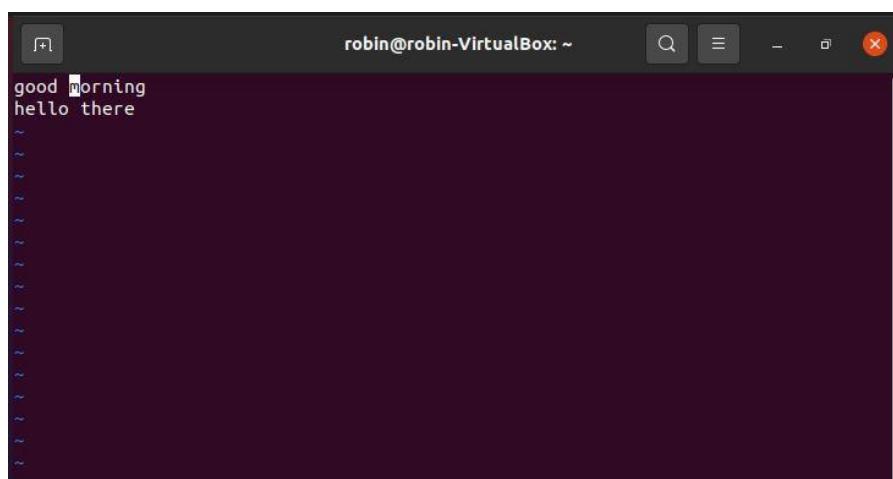
- There are mainly two modes in vi
 - Command mode
 - This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.
 - vi always starts in the command mode.
 - Input mode
 - Accessed by typing “i”
 - This mode permits insertion of new text, editing of existing text or replacement of existing text.
 - This mode enables you to insert text into the file. Everything that is typed in this mode is interpreted as input and is placed in the file.
 - To come out of the insert mode, press the Esc key, which will take you back to the command mode.
- Commands for moving within files

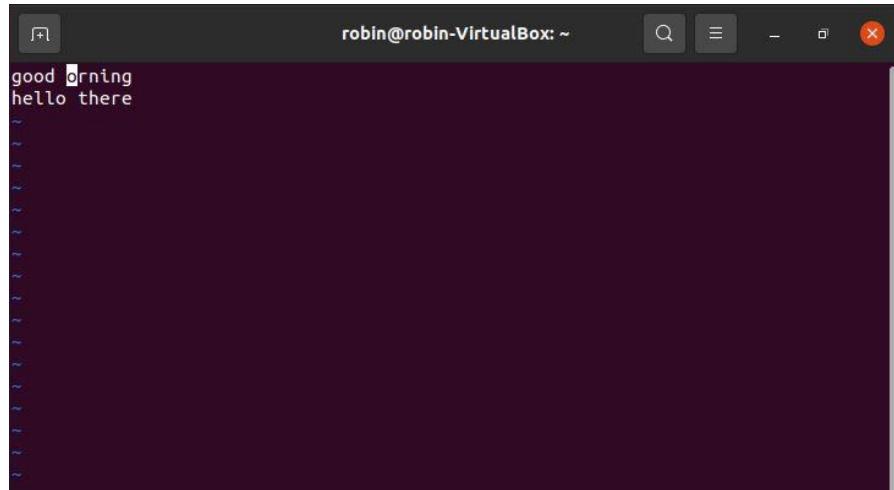
Sr.No.		Command & Description
1	k	Moves the cursor up one line
2	j	Moves the cursor down one line
3	h	Moves the cursor to the left one character position
4	l	Moves the cursor to the right one character position

- Commands for editing

Sr.No.	Command & Description
1	i Inserts text before the current cursor location
2	I Inserts text at the beginning of the current line
3	a Inserts text after the current cursor location
4	A Inserts text at the end of the current line
5	o Creates a new line for text entry below the cursor location
6	O Creates a new line for text entry above the cursor location

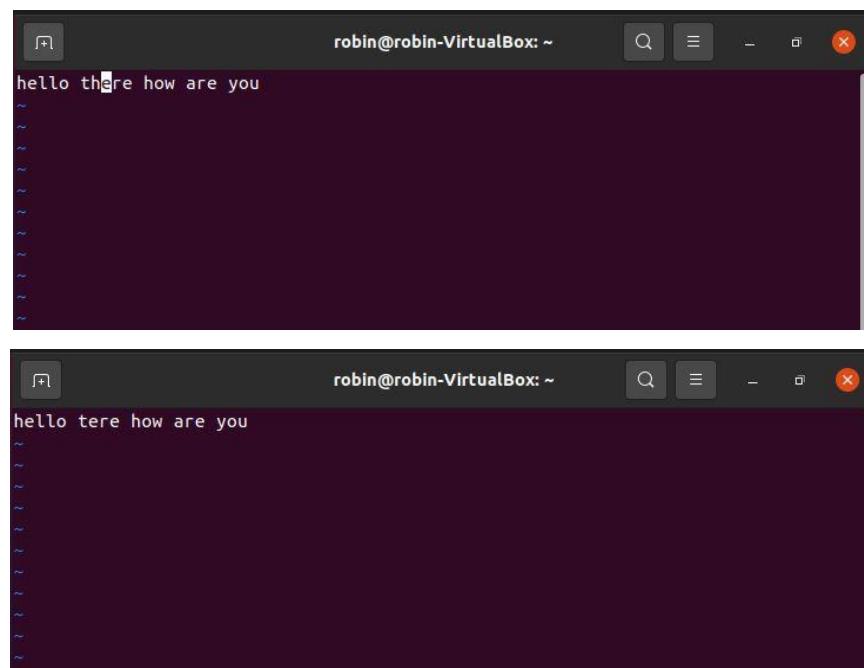
- Commands for deletions
 - **x** - Deletes the character under the cursor location





```
good morning
hello there
```

X - Deletes the character before the cursor location



```
hello there how are you
```



```
hello tere how are you
```

dw - Deletes from the current cursor location to the next word



```
Welcome
how are you?
see you later
```



```
Welcome
how you?
see you later
~
```

d^ - Deletes from the current cursor position to the beginning of the line



```
how are you?
am i distrubed you
see you later
~
```



```
~
```

d\$ - Deletes from the current cursor position to the end of the line



```
Welcome to the world
happy to see you
see you later
~
```



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~". The terminal has a dark background with white text. It displays the following text:
Welcome to the world
happy to█
see you later
~
~
~
~
~
~
~
~
The cursor is positioned at the end of the word "to".

D - Deletes from the cursor position to the end of the current line



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~". The terminal has a dark background with white text. It displays the following text:
Welcome to the world
happy to see youh
see you later
~
~
~
~
~
~
~
~
The cursor is positioned at the end of the word "youh".



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~". The terminal has a dark background with white text. It displays the following text:
Welcome to the world
happy to█
see you later
~
~
~
~
~
~
~
~
The cursor is positioned at the end of the word "to".

Dd - Deletes the line the cursor is on



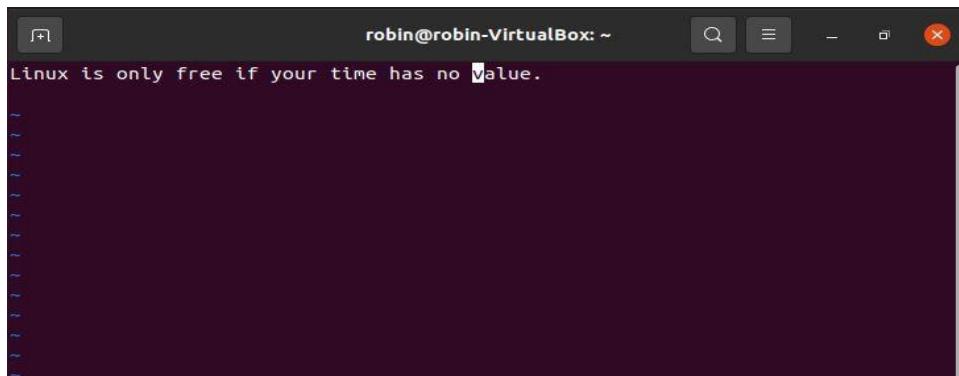
A screenshot of a terminal window titled "robin@robin-VirtualBox: ~". The terminal has a dark background with white text. It displays the following text:
Welcome to the world
hello there
isee you later
~
~
~
~
~
~
~
~
The cursor is positioned at the start of the second line.



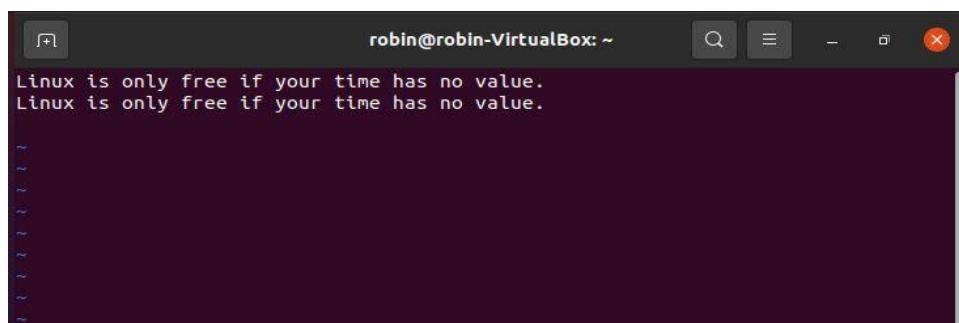
```
Welcome to the world
i see you later
```

Commands for copying and pasting

- **yy** - Copies the current line.
- **yw** - Copies the current word from the character the lowercase w cursor is on, until the end of the word.
- **p** - Puts the copied text after the cursor.



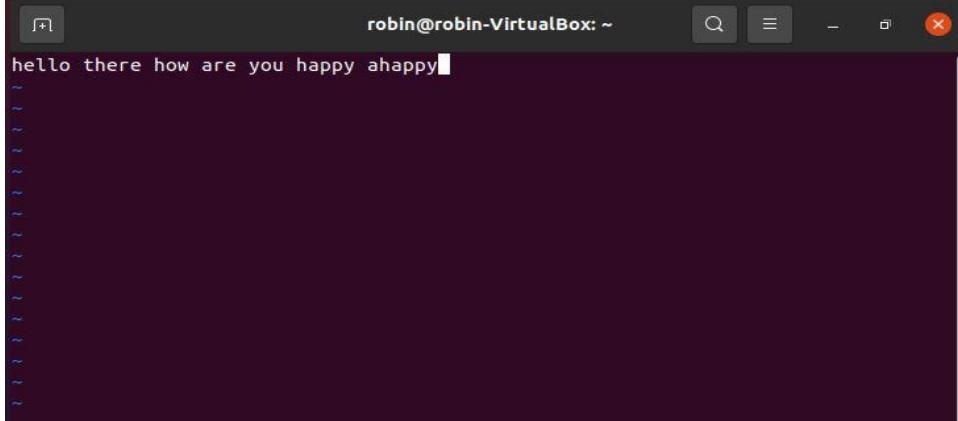
```
Linux is only free if your time has no value.
```



```
Linux is only free if your time has no value.
Linux is only free if your time has no value.
```

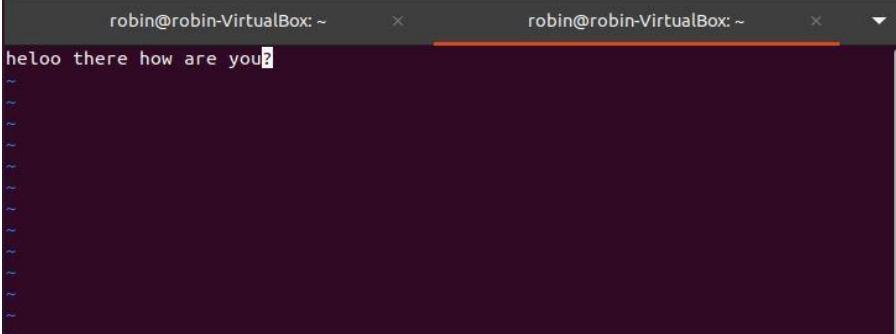


```
robin@robin-VirtualBox: ~
hello there how are you happy a
```



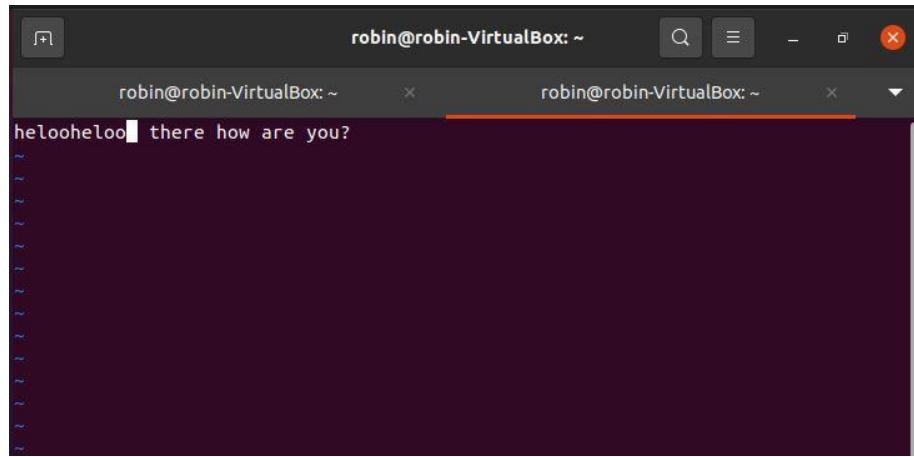
```
robin@robin-VirtualBox: ~
hello there how are you happy ahappy
```

P - Puts the yanked text before the cursor.



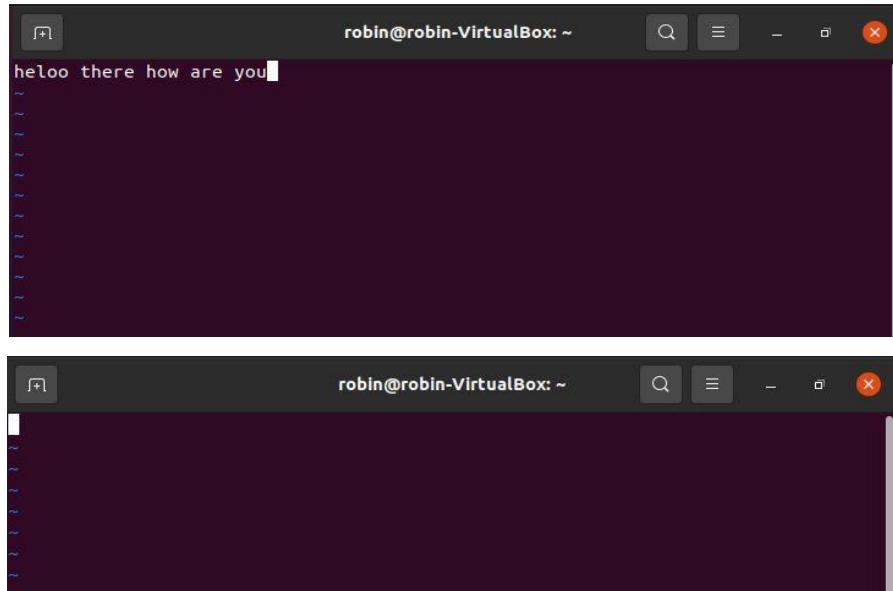
```
robin@robin-VirtualBox: ~
heloo there how are you?
```

```
robin@robin-VirtualBox: ~
```



```
robin@robin-VirtualBox: ~
robin@robin-VirtualBox: ~
heloo heloo there how are you?
```

cc - Removes the contents of the line, leaving you in insert mode.



```
robin@robin-VirtualBox: ~
robin@robin-VirtualBox: ~
heloo there how are you
he
```

- ZZ is for save and Exit (Note this command is uppercase)
- :q is to exit, if you have not made any changes to the file

EXPERIMENT NO: 11

FAMILIARIZATION WITH EMACS TEXT EDITOR

The emacs text editors are-

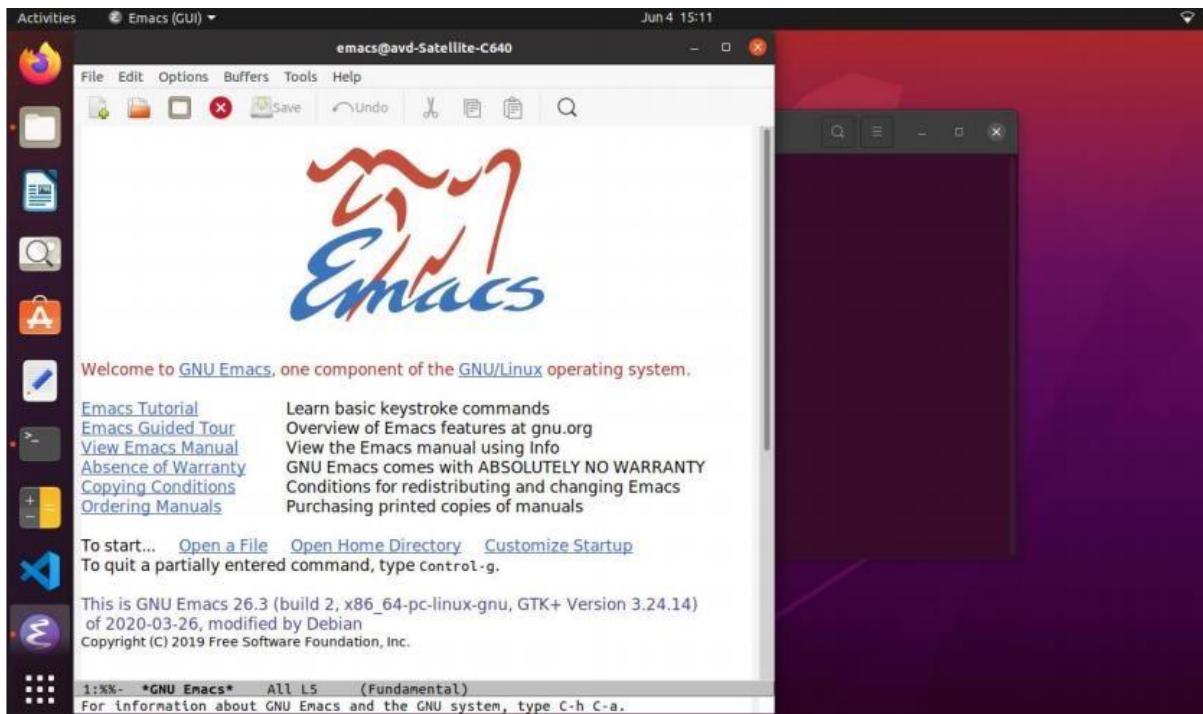
- Faster
- Powerful
- Simple
- very popular editor.
- most complex and flexible text editor than other text editors.
- freedom to the users to editing text files
- The emacs program can be running either in a terminal or in a GUI
- There are two major “brands” of emacs :GNU Emacs and Xemacs, and they have very similar functions.
- GNU Emacs –most popular version of Emacs which was created by Richard Stallman for the GNU Project.
- Xemacs is a variant that branched from GNU Emacs In 1991.
- According to Vi editor ,the emacs editor does not use an insert mode
- Commands in emacs are either control characters(hold down the <Ctrl> Key while typing another character)

To install emacs in ubuntu run the command

\$ sudo apt-get install emacs

To create a new or existing file, run

\$ emacs newfile.txt



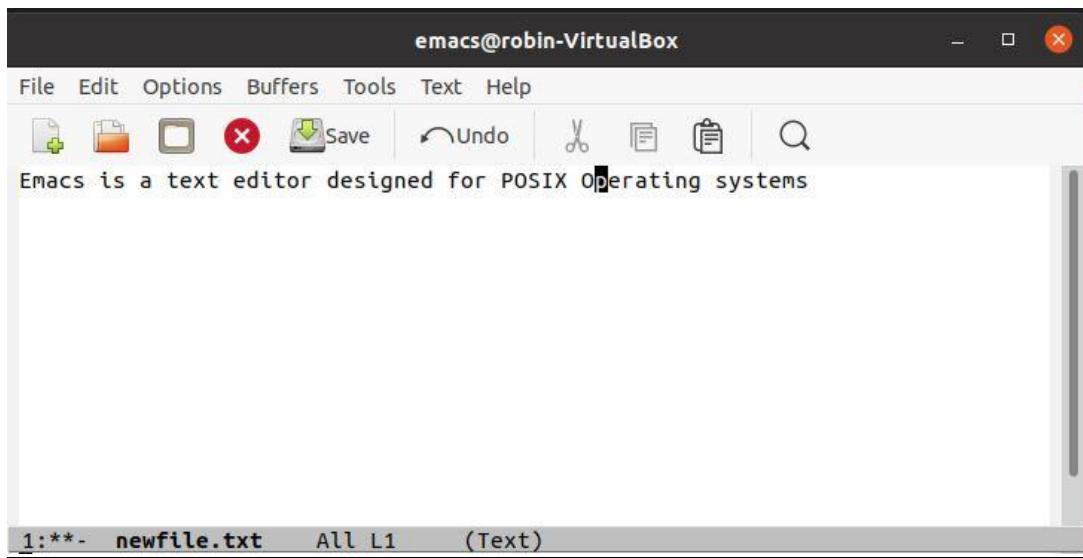
- Commands in emacs

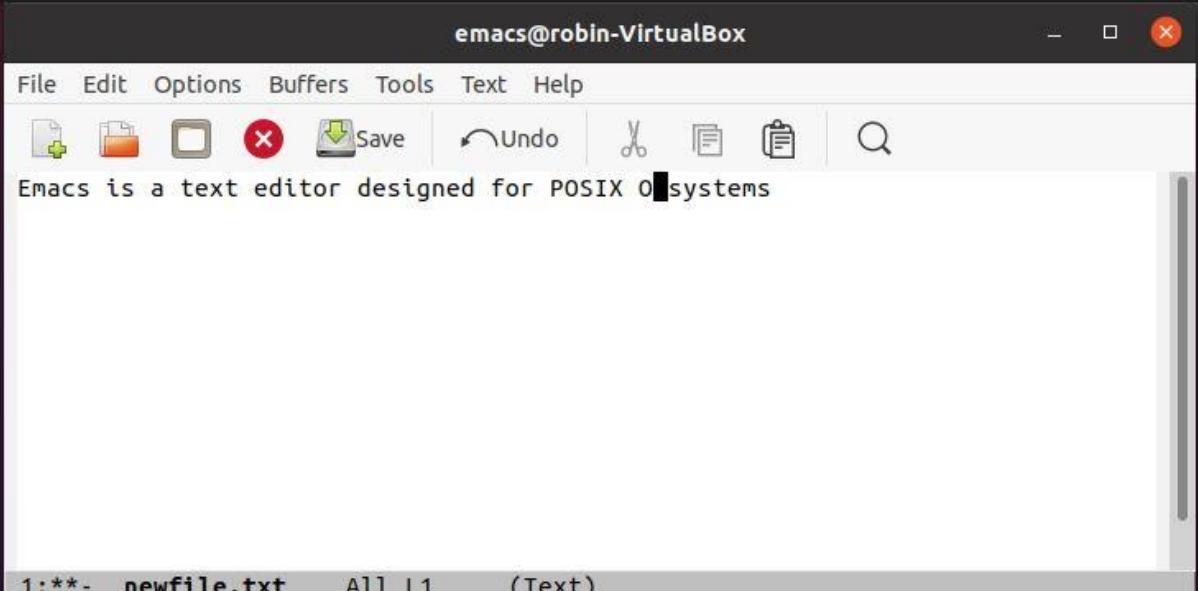
Commands	Actions
CTRL-X + CTRL-C	To exit emacs
CTRL-H	To access help documentation
CTRL-X + CTRL-S	To save the current file
CTRL-X + CTRL-W	To save as new file
CTRL-X + CTRL-U	To undo the last edit
CTRL-G	To cancel the current command
CTRL-D	To delete(Kill) the character at the cursor
ESC -D	To delete characters from the cursor to the end of the current line

Commands	Actions
CTRL-K	To delete characters from the cursor to the end of the current line
ESC -W	To copy region as kill
CTRL-Y	To paste into the buffer what had been deleted
CTRL-S	To search the string from the current cursor position
CTRL-R	To search the string above the current cursor position

Commands	Action
ESC-F	To move cursor forward
ESC-B	To move the cursor backward one word at a time
CTRL-A	To move the cursor to the beginning of the current line
CTRL-E	To move the cursor to the end of the current line
ESC-<	To move the cursor beginning of the buffer
ESC->	To move the cursor to the end of the buffer

CTRL-D - To delete(Kill) the character at the cursor





emacs@robin-VirtualBox

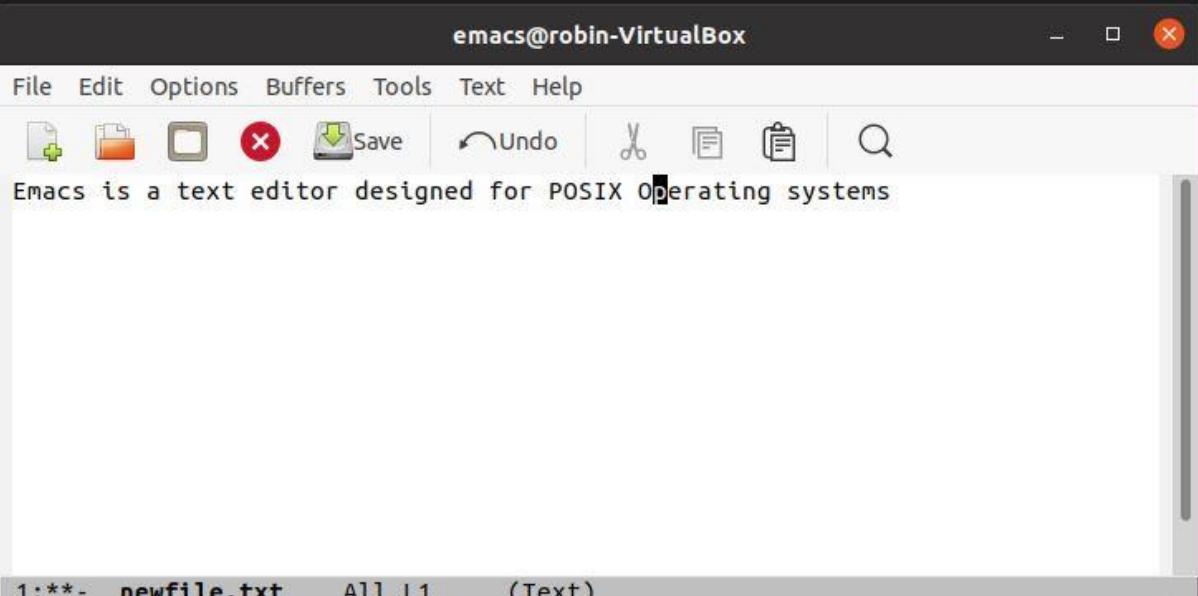
File Edit Options Buffers Tools Text Help

Save Undo

Emacs is a text editor designed for POSIX 0systems

1:***- newfile.txt All L1 (Text)

ESC -D- To delete characters from the cursor to the end of the current line



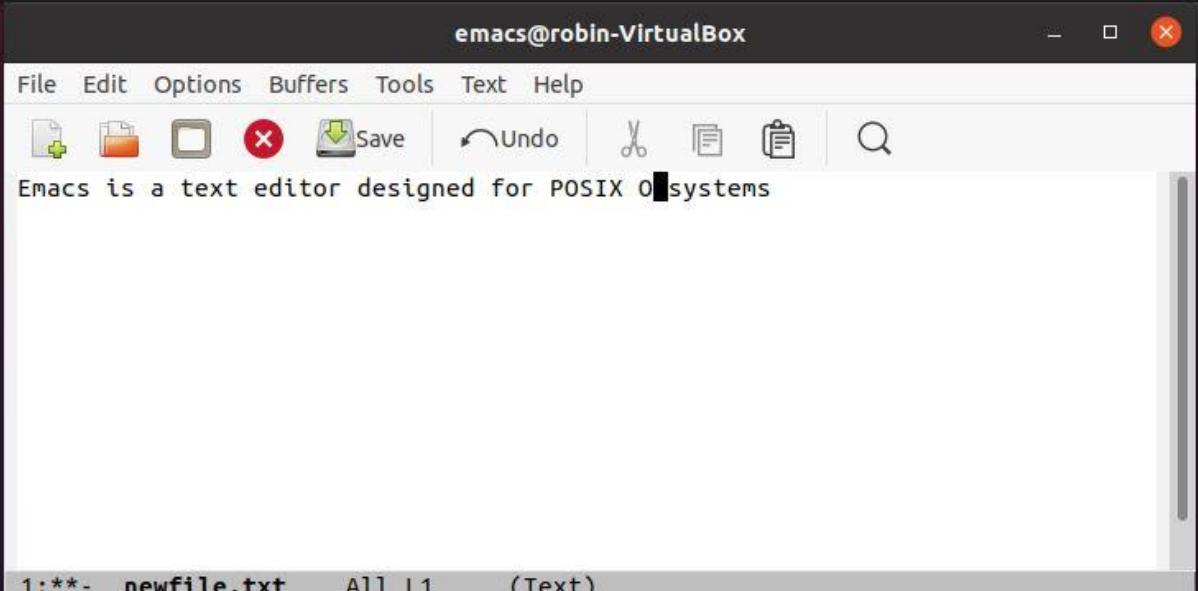
emacs@robin-VirtualBox

File Edit Options Buffers Tools Text Help

Save Undo

Emacs is a text editor designed for POSIX Operating systems

1:***- newfile.txt All L1 (Text)



emacs@robin-VirtualBox

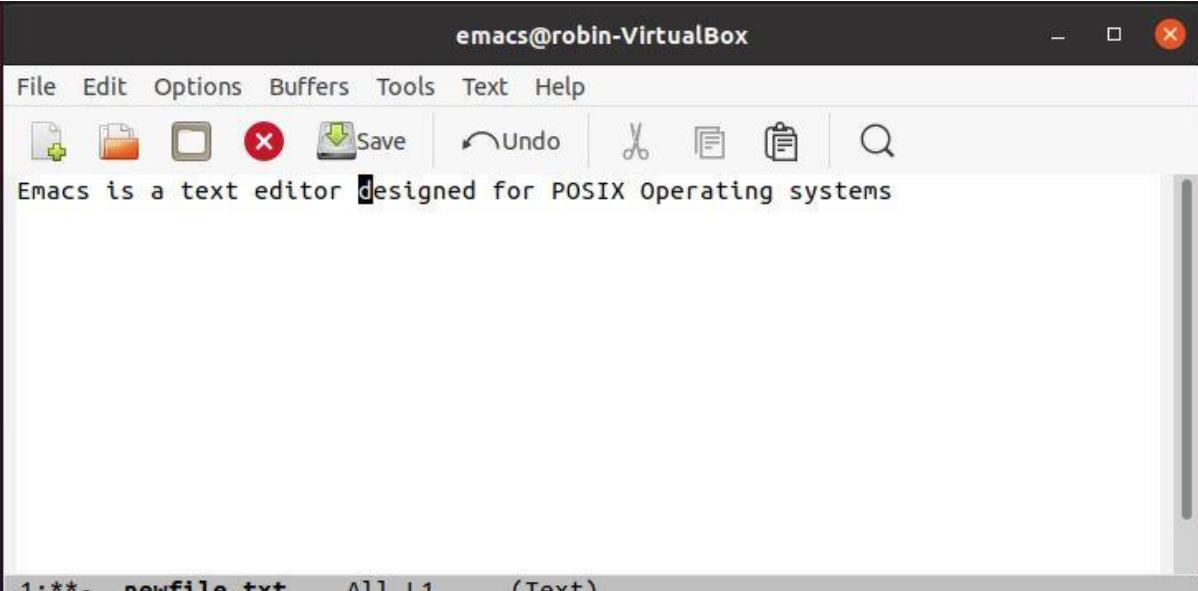
File Edit Options Buffers Tools Text Help

Save Undo

Emacs is a text editor designed for POSIX 0systems

1:***- newfile.txt All L1 (Text)

CTRL K - To delete characters from the cursor to the end of the current line



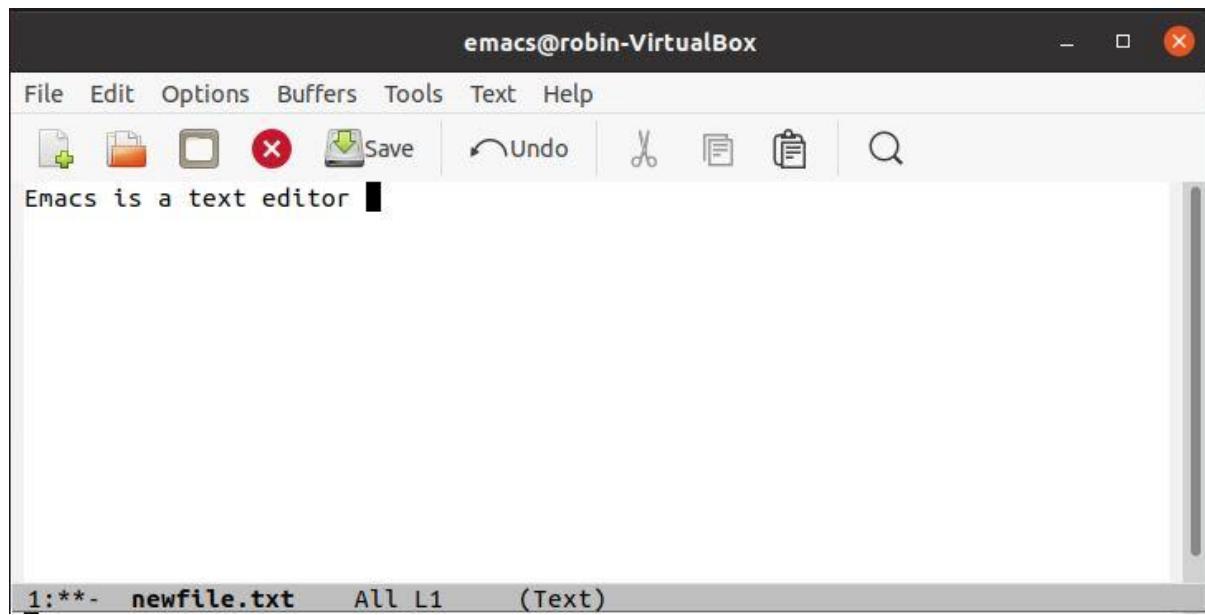
emacs@robin-VirtualBox

File Edit Options Buffers Tools Text Help

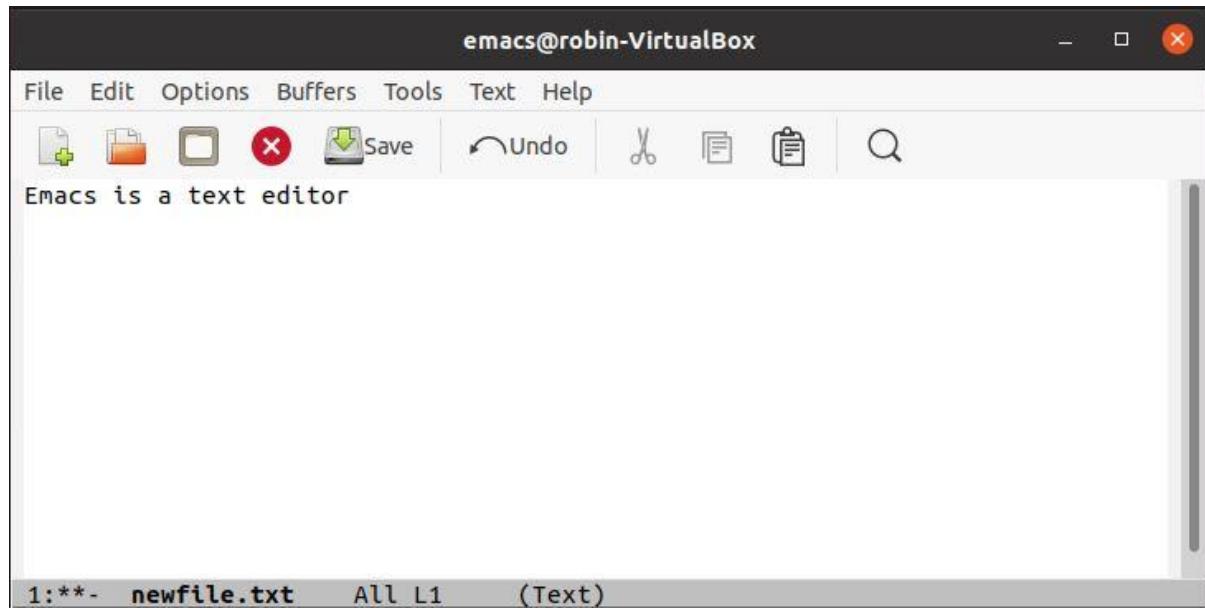
Save Undo

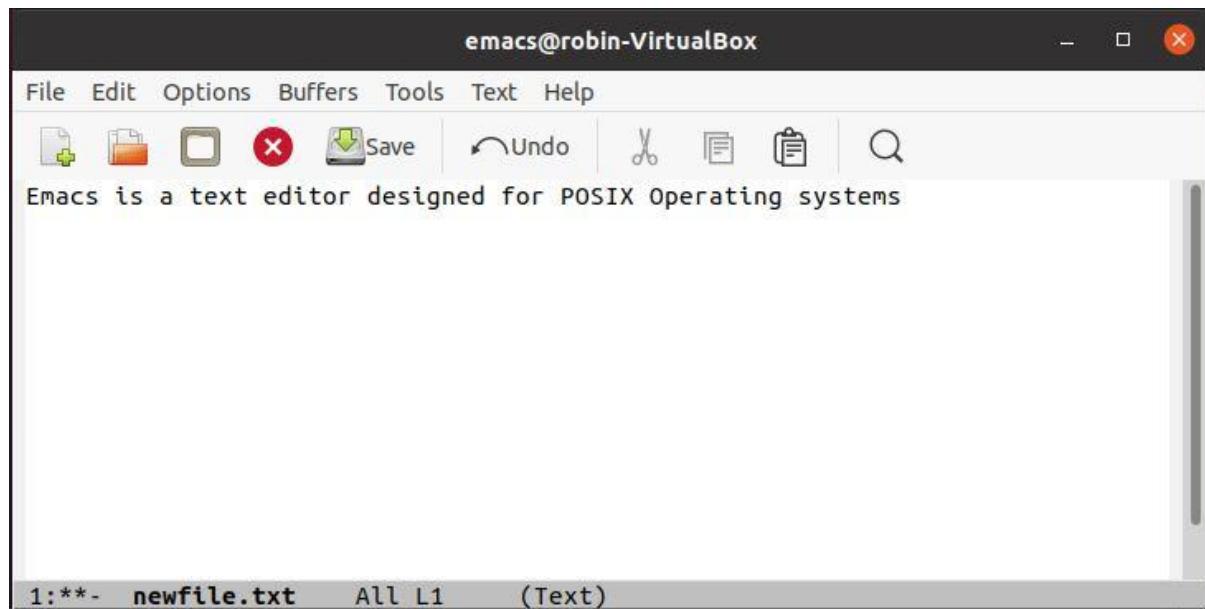
Emacs is a text editor designed for POSIX Operating systems

1:***- newfile.txt All L1 (Text)

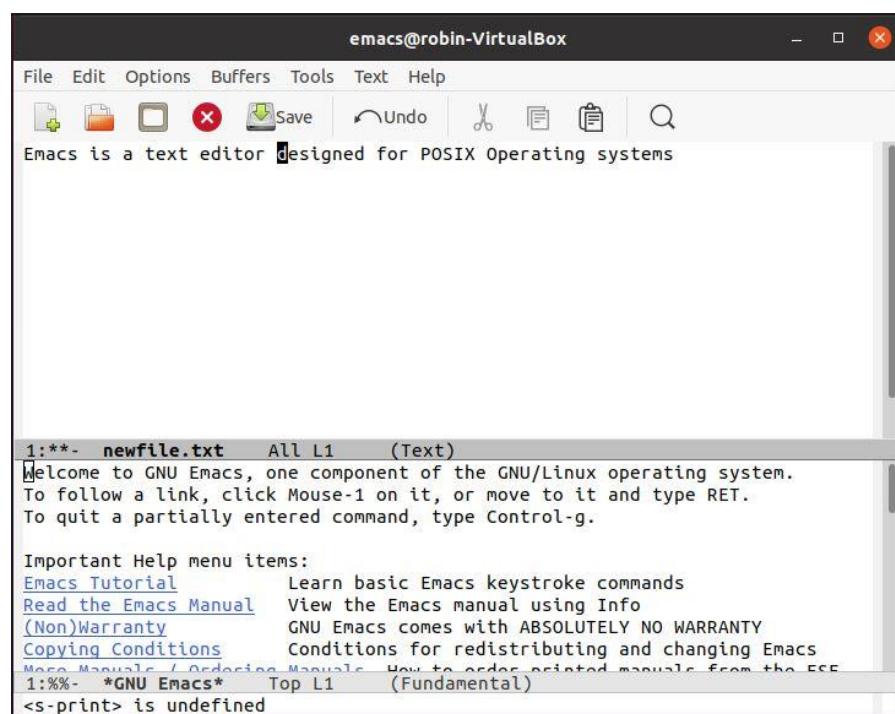


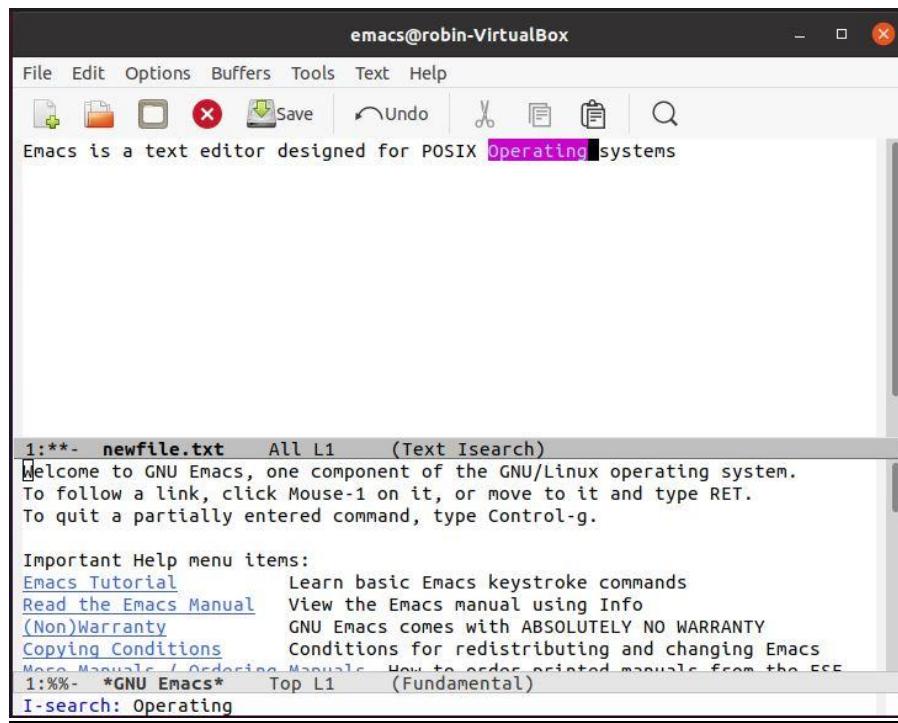
CTRL Y - To paste into the buffer what had been deleted





CTRL S - To search the string from the current cursor position





The screenshot shows the Emacs text editor window titled "emacs@robin-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Text", and "Help". Below the menu bar is a toolbar with icons for file operations like Open, Save, Undo, and Cut/Paste. The main buffer area displays the text:

```
Emacs is a text editor designed for POSIX Operating systems
```

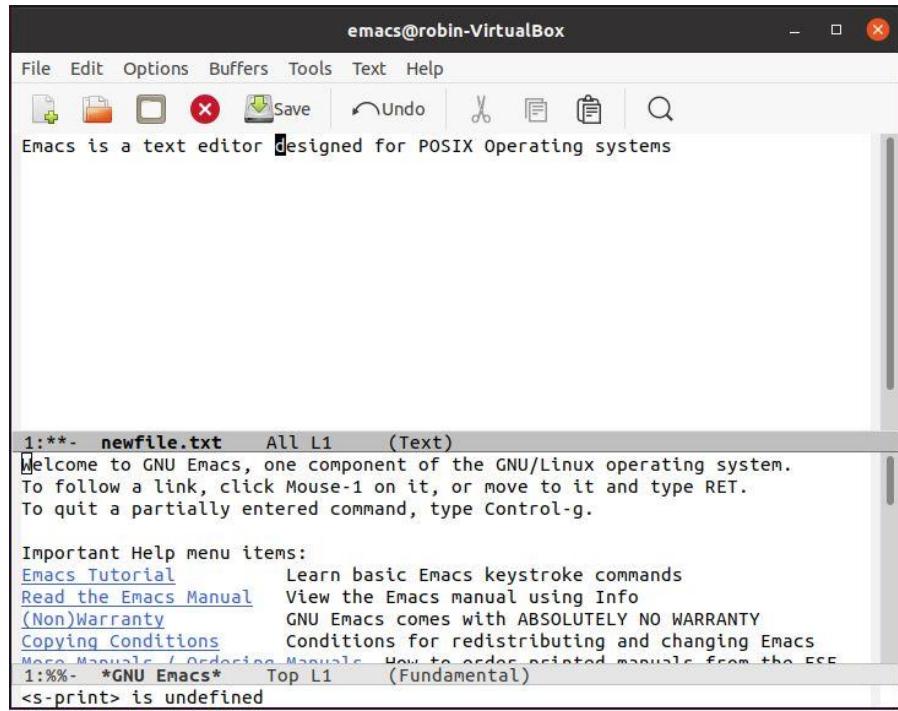
Below this, the status bar shows the current buffer and mode:

```
1:***- newfile.txt All L1 (Text Isearch)
```

The "Help" menu is open, showing the following items:

- [Emacs Tutorial](#) Learn basic Emacs keystroke commands
- [Read the Emacs Manual](#) View the Emacs manual using Info
- [\(Non\)Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
- [Copying Conditions](#) Conditions for redistributing and changing Emacs
- [Manuals / Ordering Manuals](#) How to order printed manuals from the FSF
- [1:%%- *GNU Emacs*](#) Top L1 (Fundamental)
- I-search: Operating

CTRL R - To search the string above the current cursor position



The screenshot shows the Emacs text editor window titled "emacs@robin-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Text", and "Help". Below the menu bar is a toolbar with icons for file operations like Open, Save, Undo, and Cut/Paste. The main buffer area displays the text:

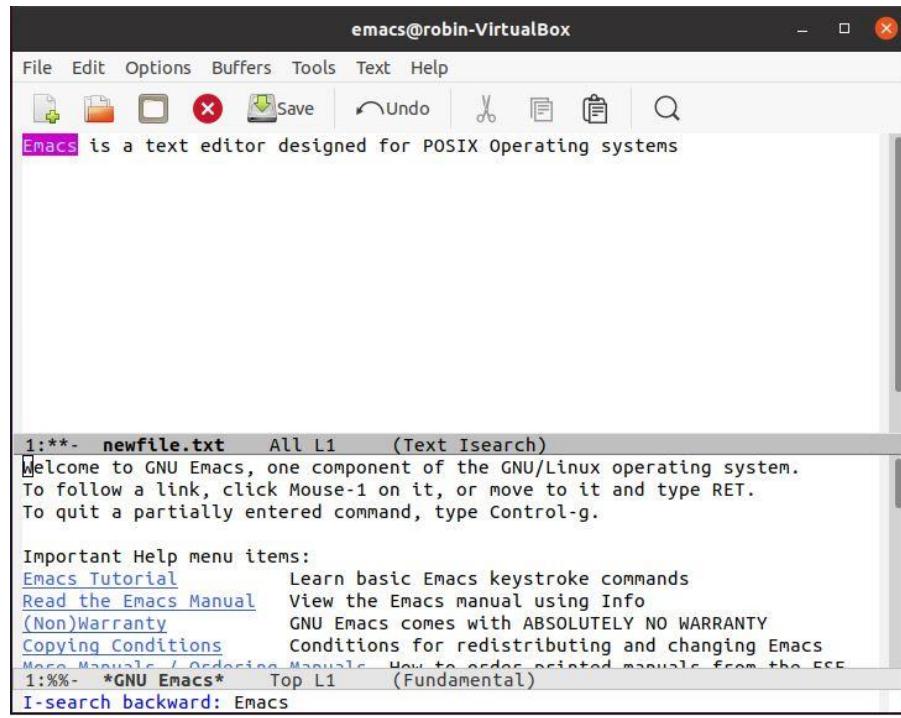
```
Emacs is a text editor designed for POSIX Operating systems
```

Below this, the status bar shows the current buffer and mode:

```
1:***- newfile.txt All L1 (Text)
```

The "Help" menu is open, showing the following items:

- [Emacs Tutorial](#) Learn basic Emacs keystroke commands
- [Read the Emacs Manual](#) View the Emacs manual using Info
- [\(Non\)Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
- [Copying Conditions](#) Conditions for redistributing and changing Emacs
- [Manuals / Ordering Manuals](#) How to order printed manuals from the FSF
- [1:%%- *GNU Emacs*](#) Top L1 (Fundamental)
- <s-print> is undefined



- The emacs editor provides environment to programmers develop program such as c, java, HTML and shell script etc.
- After write the program source code, it also compile, link, debug and execute itself in the terminal window to test it.

EXPERIMENT NO: 12

LINUX FILE SYSTEM

Linux File System

A Linux file system is a structured collection of files on a disk drive or a partition. A partition is a segment of memory and contains some specific data. In our machine, there can be various partitions of the memory. Generally, every partition contains a file system.

The Linux file system contains the following sections:

- The root directory (/)
- A specific data storage format (EXT3, EXT4, BTRFS, XFS and so on)
- A partition or logical volume having a particular file system.

Some key features of Linux file system are as following:

- Specifying paths (eg: /home/ My Document/ Work)
- Partition, Directories, and Drives
- Case Sensitivity
- File Extensions
- Hidden files

Types of Linux file system

When we install the Linux operating system, Linux offers many file systems such as Ext, Ext2, Ext3, Ext4, JFS, ReiserFS, XFS, btrfs, and swap.

1. Ext, Ext2, Ext3 and Ext4 file system

The file system Ext stands for Extended File System. It was primarily developed for MINIX OS. The Ext file system is an older version, and is no longer used due to some limitations.

Ext2 is the first Linux file system that allows managing two terabytes of data. Ext3 is developed through Ext2; it is an upgraded version of Ext2 and contains backward compatibility. The major drawback of Ext3 is that it does not support servers because this file system does not support file recovery and disk snapshot.

Ext4 file system is the fastest file system among all the Ext file systems. It is a very compatible option for the SSD (solid-state drive) disks, and it is the default file system in Linux distribution.

2. JFS File System

JFS stands for Journalized File System, and it is developed by IBM for AIX Unix. It is an alternative to the Ext file system. It can also be used in place of Ext4, where stability is needed with few resources. It is a handy file system when CPU power is limited.

3. ReiserFS File System

ReiserFS is an alternative to the Ext3 file system. It has improved performance and advanced features. In the earlier time, the ReiserFS was used as the default file system in SUSE Linux, but later it changed some policies, so SUSE returned to Ext3. This file system dynamically supports the file extension, but it has some drawbacks in performance.

4. XFS File System XFS file system was considered as high-speed JFS, which is developed for parallel I/O processing. NASA is still using this file system with its high storage server (300+ Terabyte server).
5. Btrfs File System Btrfs stands for the B tree file system. It is used for fault tolerance, repair system, fun administration, extensive storage configuration, and more. It is not a good suit for the production system.
6. Swap File System The swap file system is used for memory paging in Linux operating system during the system hibernation. A system that never goes in hibernate state is required to have swap

space equal to its RAM size. It makes sense to explore the Linux filesystem from a terminal window, In fact, that is the name of the first tool you'll install to help you on the way: tree. sudo apt install tree

7. To open the tree simply type tree in the terminal

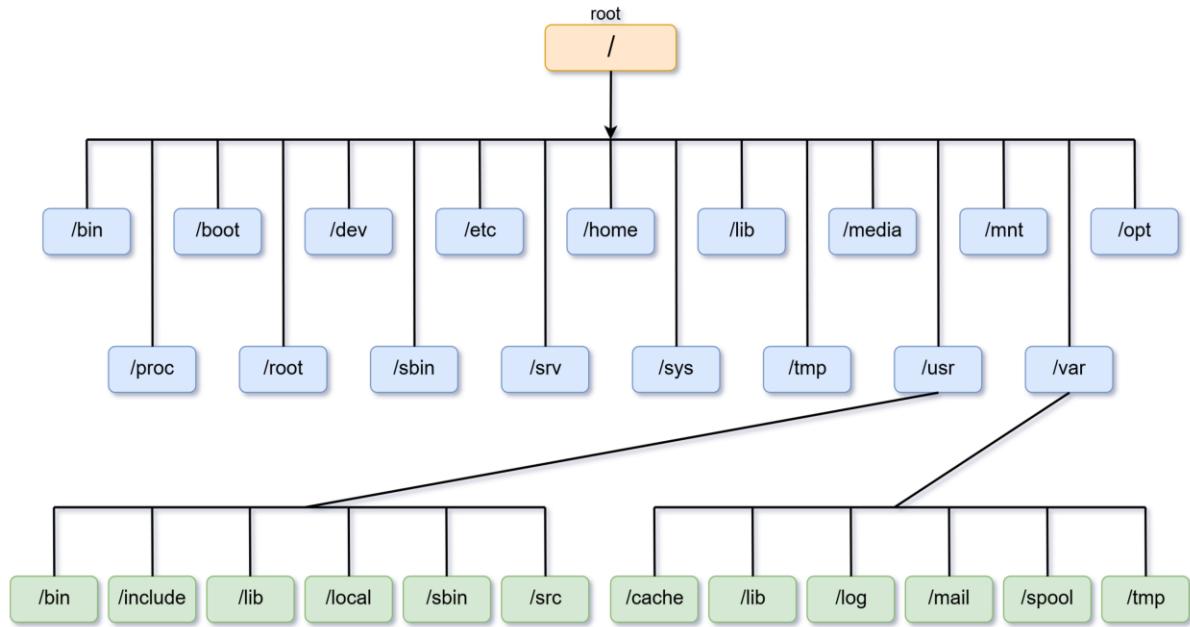
```
robin@robin-VirtualBox: ~ $ tree
.
├── Desktop
│   └── empdict
├── Documents
├── Downloads
├── index.html
├── index.html.1
├── Music
│   ├── file.txt
│   └── test
│       └── computer.txt
│           └── test.tar
├── #newfile.txt#
├── Pictures
│   └── Screenshot from 2021-09-08 20-51-08.png
├── Public
└── snap
    └── snap-store
        ├── 481
        ├── 547
        └── common
            └── current -> 547
└── Templates
└── Videos

16 directories, 7 files
robin@robin-VirtualBox: $
```

File System Hierarchy

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.
- Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS and are not considered authoritative for platforms other than Linux.



The Root Directory

All the directories in the Linux system come under the root directory which is represented by a forward slash (/). Everything in your system can be found under this root directory even if they are stored in different virtual or physical devices.

```

robin@robin-VirtualBox:~$ ls /
bin   dev   lib   libx32   mnt   root   snap   sys   var
boot  etc   lib32  lost+found  opt   run   srv   tmp
cdrom home  lib64  media    proc   sbin  swapfile  usr
robin@robin-VirtualBox:~$ 
  
```

Binary Directory

Binary files are the files which contain compiled source code (or machine code). They are also called executable files because they can be executed on the computer.

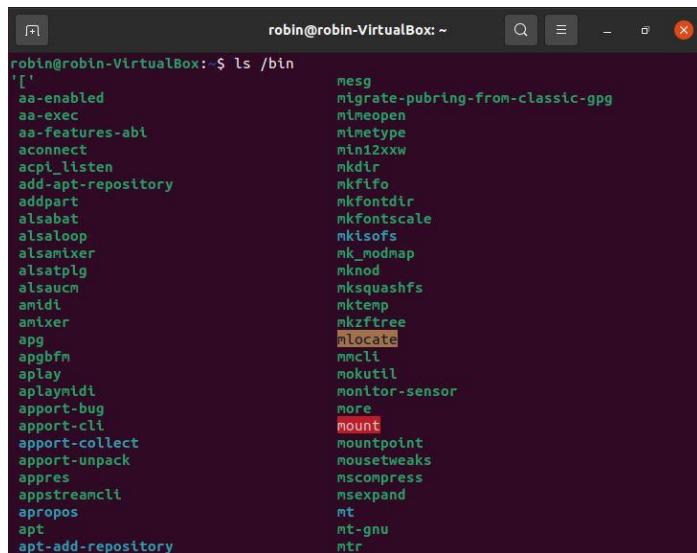
Binary directory contains following directories:

- **/bin**
- **/sbin**
- **/lib**
- **/opt**

/bin

The '/bin' directory contains user binaries, executable files, Linux commands that are used in single user mode, and common commands that are used by all the users, like cat, cp, cd, ls, etc.

The '/bin' directory doesn't contain directories.



```
robin@robin-VirtualBox:~$ ls /bin
['[', 'aa-enabled', 'aa-exec', 'aa-features-abi', 'aconnect', 'acpi_listen', 'add-apt-repository', 'addpart', 'alsabat', 'alsaloop', 'alsamixer', 'alsaplug', 'alsaucm', 'amidi', 'amixer', 'apg', 'apgbfm', 'aplay', 'aplaymidi', 'apport-bug', 'apport-cli', 'apport-collect', 'apport-unpack', 'apres', 'appstreamcli', 'apropos', 'apt', 'apt-add-repository', 'mesg', 'migrate-pubring-from-classic-gpg', 'mineopen', 'minetype', 'min12xxw', 'mkdir', 'mkfifo', 'mkfontdir', 'mkfontscale', 'mkisofs', 'mk_modmap', 'mknod', 'mksquashfs', 'mktemp', 'mkzftree', 'mlocate', 'mmccli', 'mokutil', 'monitor-sensor', 'more', 'mount', 'mountpoint', 'mouse tweaks', 'nscompress', 'msexpand', 'mt', 'mt-gnu', 'mtr']
```

The command "**ls /bin**" displays the list of '/bin' directory. Commands like cp, cat, less, touch, etc can be seen in the snapshot.

/sbin

The '/sbin' directory also contains executable files, but unlike '/bin' it only contains system binaries which require root privilege to perform certain tasks and are helpful for system maintenance purposes. e.g. fsck, root, init, ifconfig, etc.

```
robin@robin-VirtualBox: ~ $ ls /sbin
aa-remove-unknown      grpconv          pivot_root
aa-status               grpunconv         plipconfig
aa-teardown             grub-bios-setup  plymouthd
accessdb                grub-install       poweroff
acpid                   grub-macbles     pppd
adcli                   grub-mkconfig    pppdump
addgroup                grub-mkconfigg   pppoe-discovery
adduser                 grub-mkdevicemap  pppstats
agetty                  grub-probe       pptp
alsabat-test            grub-reboot     pptpsetup
alsactl                 grub-set-default pwck
alsainfo                halt             pwconv
anacron                 hparm            pwunconv
apparmor_parser         hwclock          rarp
apparmor_status          ifconfig          raw
applynupgdefaults       iiconvconfig    readprofile
aptd                    iio-sensor-proxy realm
arp                     init              reboot
arpd                   insmod            regdbdump
arptables               installkernel  remove-default-ispell
arptables-nft           install-sgmlcatalog remove-default-wordlist
arptables-nft-restore   invoke-rc.d    remove-shell
arptables-nft-save     ip                resize2fs
arptables-restore       ip6tables        rfkill
arptables-save          ip6tables-apply  rmmod
arptables-restore       ip6tables-legacy  rmt
arptables-save          ip6tables-legacy-restore rmt-tar
arptables-save          ip6tables-legacy-save route
```

The command "**ls /sbin**" displays the list of '/sbin' directory.

/lib

The '/lib' directory contains **shared libraries** which are often used by the '/bin' and '/sbin' directories. It also contains kernel module. These filenames are identifiable as ld* or lib*.so.*. For example, ld-linux.so.2 and libfuse.so.2.8.6

```
robin@robin-VirtualBox: ~ ls /lib
'${DEB_HOST_MULTIARCH}'
accountsservice
apg
apparmor
apt
aspell
binfmt.d
bluetooth
brlty
cnf-update-db
command-not-found
console-setup
cpp
crda
cups
dbus-1.0
debug
dpkg
emacs
emacsen-common
environment.d
evolution-data-server
file
firefox
firefox-addons
firewalld
firmware
gcc
locale
lp_solve
lsb
man-db
mentest86+
mine
modprobe.d
modules
modules-load.d
netplan
networkd-dispatcher
NetworkManager
nvidia
openssh
open-vm-tools
openvpn
os-prober
os-probes
os-release
packagekit
pcmciautils
pkconfig
pkg-config.multiarch
pm-utils
policykit-1
pppd
pulse-13.99.2
python2.7
```

The command "ls /lib" displays the list of '/lib' directory.

- **/lib/modules**: The '/lib/modules' stores kernel modules and has a directory for each installed kernel. Modules are meant to use extra hardware support without making a new kernel.
 - **/lib32 and /lib64**: During compilation time of libraries you'll encounter through the directories named '/lib32' and '/lib64' which will clarify register size to be used. A 64-bit system may have compatibility for 32-bit binary.

/opt

The term '**opt**' is short for optional. Its main purpose is to store optional application software packages. Add-on applications from individual vendors should be installed in '/opt'. And so in some systems '/opt' is empty as they may not have any add-on application.

Configuration Directory

The configuration directory contains configured files which configure the parameters and initial settings for some computer programs.

Configuration directory have following subdirectories:

- **/boot**
- **/etc**

/boot

The '/boot' directory contains boot loader files which are essential to boot the system. In other words, they only contain files which are needed for a basic Linux system to get up and going.

You may find the '/boot/grub' directory which contains '/boot/grub/grub.cfg' (older systems may have /boot/grub/grub.conf) which defines the boot menu that is displayed before the kernel starts.

```
robin@robin-VirtualBox:~$ ls /boot
config-5.8.0-25-generic      memtest86+.elf
config-5.8.0-63-generic      memtest86+_multiboot.bin
efi                           System.map-5.8.0-25-generic
grub                          System.map-5.8.0-63-generic
initrd.img                     vmlinuz
initrd.img-5.8.0-25-generic   vmlinuz-5.8.0-25-generic
initrd.img-5.8.0-63-generic   vmlinuz-5.8.0-63-generic
initrd.img.old                 vmlinuz.old
memtest86+.bin
memtest86+.elf
memtest86+_multiboot.bin
System.map-5.8.0-25-generic
System.map-5.8.0-63-generic
vmlinuz
vmlinuz-5.8.0-25-generic
vmlinuz-5.8.0-63-generic
vmlinuz.old
robin@robin-VirtualBox:~$
```

The command "**ls /boot**" displays the list of '/boot' directory.

/etc

All the machine related configuration files are kept in '/etc'. Almost everything related to the configuration of your system is placed here. It also contains startup and shutdown shell scripts which are used to start and stop a program. All the files are static and text based and no binary files can be placed in this directory.

The meaning of 'etc' is very controversial. Earlier it was referred to as '**Etcetera**' because it could contain all the files that did not belong from anywhere else. But recently its most likely meaning is '**Editable Text Configuration**' or '**Extended Tool chest**'.

Configuration files will have an extension **.conf**.

```
robin@robin-VirtualBox: $ ls /etc
acpi           hosts          profile.d
adduser.conf   hosts.allow    protocols
alsa           hosts.deny    pulse
alternatives   hp             python3
anacrontab     ifplugd       python3.8
apg.conf       ImageMagick-6 rc0.d
apm            init           rc1.d
apparmor       init.d        rc2.d
apparmor.d     inittab       rc3.d
apport         inputrc       rc4.d
appstream.conf insserv.conf  rc5.d
apt            ipp-usb       rc6.d
avahi          iproute2     rcS.d
bash.bashrc    issue         resolv.conf
bash_completion kernel        rmt
bash_completion.d kernel-img.conf
bindresvport.blacklist kerneloops.conf
blinfmt.d      ldap           rpc
bluetooth     ld.so.cache   rsyslog.conf
brlapi.key     ld.so.conf    rsyslog.d
brltty         ld.so.conf.d rygel.conf
brltty.conf    legal          sane.d
ca-certificates libao.conf   security
ca-certificates.conf libaudit.conf selinux
chatscripts    libblockdev   sensors3.conf
console-setup   libnl-3       sensors.d
cracklib      libpaper.d   services
cron.d         libpaper.d   sgml
shadow         shadow        shadow-
```

The command "**ls /etc**" displays the list of '/etc' directory.

Some common directories are:

- **/etc/init.d/**: The term 'init' is short for **initialization**. This directory contains script to control the system or to start and stop the daemons (background process). The 'init' is a daemon process that continues running until the system is shut down.
- **/etc/X11/**: The X Window system configuration files are stored in this directory. The configuration file of graphical display (xorg.conf) is also stored here.
- **/etc/skel/**: The term 'skel' is short for **skeleton**. Everything in the system has a skeleton which is called hidden file and is stored in this directory. It is not an important part in the system and can be deleted but still it serves a specific purpose. Its purpose is to serve the basic set of files, a basic framework which can be used in the creation of a new user.

Data Directory

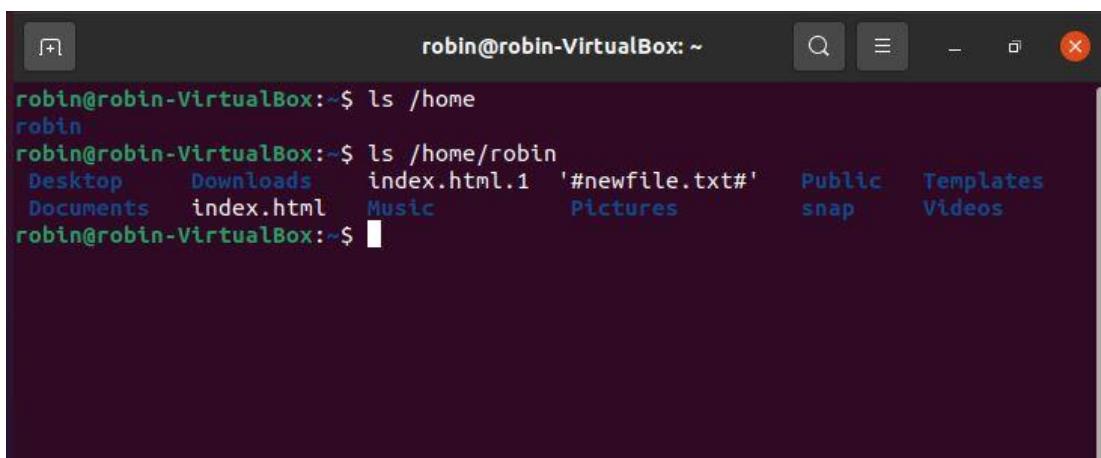
Data directory is used to store data of the system.

Data directory contains following directories.

- /home
- /root
- /srv
- /media
- /mnt
- /tmp

/home

The '/home' directory stores users personnel files. After the '/home' there is a directory which is generally named at the user's name like we have '**/home/abcd**'. Inside this directory we have our sub-directories like Desktop, Downloads, Documents, pictures, etc.



A screenshot of a terminal window titled 'robin@robin-VirtualBox: ~'. The terminal shows two commands: 'ls /home' and 'ls /home/robin'. The first command lists the root directory '/home' containing the user 'robin'. The second command lists the contents of the user's home directory '/home/robin', which includes sub-directories like Desktop, Downloads, Documents, index.html, Music, Pictures, Public, Templates, snap, and Videos, along with files like index.html.1 and '#newfile.txt#'. The terminal has a dark background with light-colored text and standard Linux-style icons.

```
robin@robin-VirtualBox:~$ ls /home
robin
robin@robin-VirtualBox:~$ ls /home/robin
Desktop    Downloads   index.html.1 '#newfile.txt#'  Public   Templates
Documents  index.html  Music          Pictures      snap     Videos
robin@robin-VirtualBox:~$
```

The command "**ls /home**" displays '**abcd**'. While command "**ls /home/abcd**" displays '**abcd**' sub-directories.

Note: Tilde (~) sign indicates "**/home/abcd**". For example, if you want to give the command "**/home/abcd/Desktop**" so instead of writing this you can also write "**~/Desktop**", both are the same.

/root

The '/root' directory is the home directory of the root user.

Please note that '/root' directory is different from (/) root.

/srv

The term 'srv' is short for **service**. The '/srv' directory contains server specific data for services provided by the system like www, cvs, rysync, ftp, etc.

/media

The '/media' directory acts as a mount point for removable media devices such as CD-Rom, floppy, USB devices, etc.

This is newly introduced directory and hence a system can run without this directory also.

The command "**ls /media**" displays '/media' content.

/mnt

The term 'mnt' stands for **mount**. The '/mnt' directory should be empty and sysadmins can only mount temporary filesystems.

/tmp

The term 'tmp' stands for **temporary**. Data stored in '/tmp' is temporary and may use either disk space or RAM. When system is rebooted, files under this directory are automatically deleted. So it is advisable that never use '/tmp' to store important data.

File Permissions

The concept of Linux File permission and ownership is crucial in Linux. Here, we will explain Linux permissions and ownership and will discuss both of them. Let us start with the Ownership.

Ownership of Linux files

Every file and directory on your Unix/Linux system is assigned 3 types of owner, given below.

User

A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

Group

A user- group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only this group members and no one else can read or modify the files.

Other

Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

Let us understand the **Permission system** on Linux.

Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- **Execute:** In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

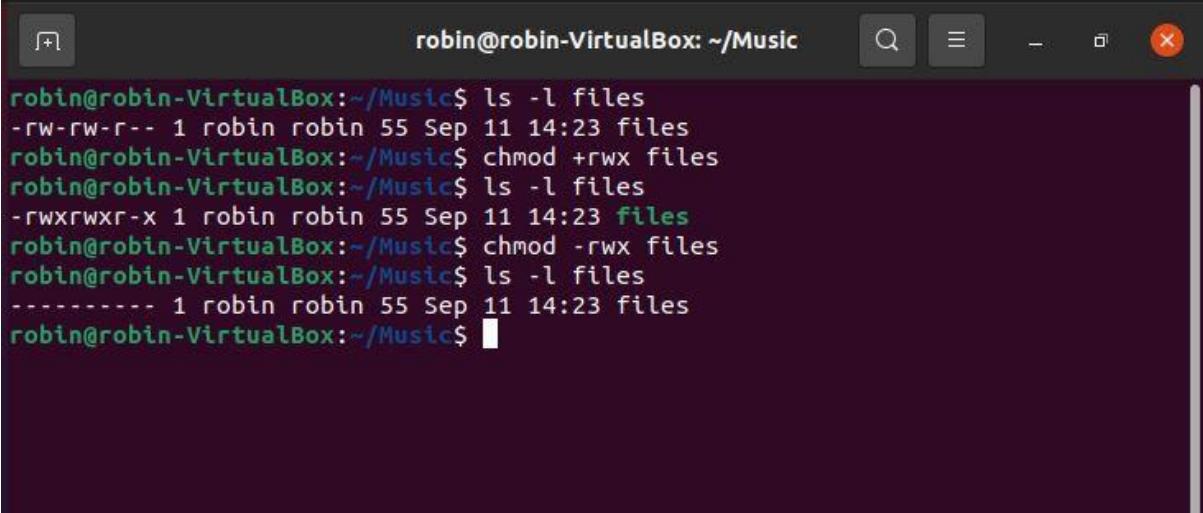
You can change the permissions with chmod command accordingly to your need. Below are some examples to change the permissions for different groups.

To add permissions to a group.

chmod +rwx filename

To remove permissions from a group.

chmod -rwx filename



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~/Music". The terminal shows the following command sequence:

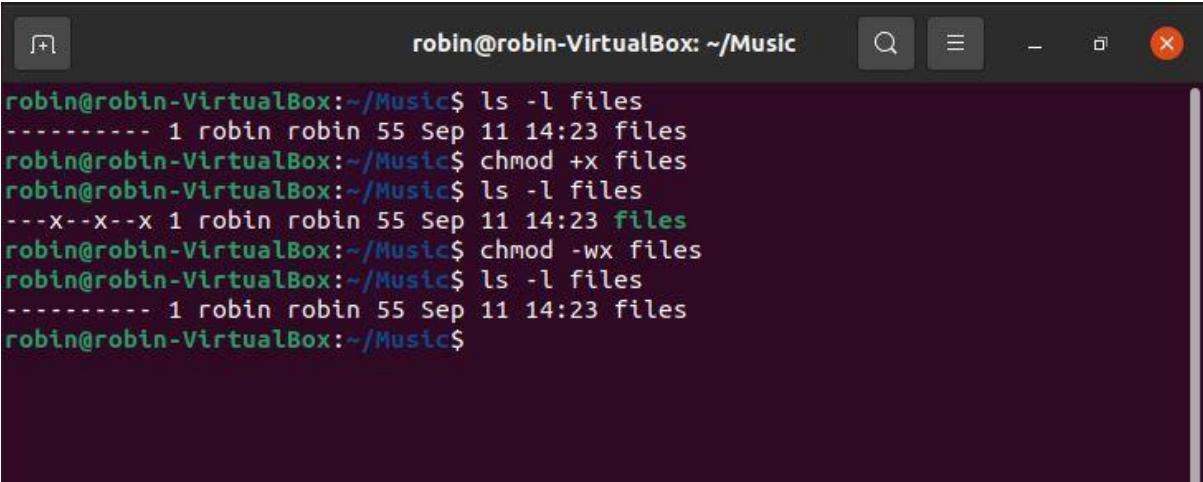
```
robin@robin-VirtualBox:~/Music$ ls -l files
-rw-rw-r-- 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$ chmod +rwx files
robin@robin-VirtualBox:~/Music$ ls -l files
-rwxrwxr-x 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$ chmod -rwx files
robin@robin-VirtualBox:~/Music$ ls -l files
----- 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$
```

To allow executable permissions.

chmod +x filename

To take out write and executable permissions.

chmod -wx filename



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~/Music". The terminal shows the following command sequence:

```
robin@robin-VirtualBox:~/Music$ ls -l files
----- 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$ chmod +x files
robin@robin-VirtualBox:~/Music$ ls -l files
---x---x 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$ chmod -wx files
robin@robin-VirtualBox:~/Music$ ls -l files
----- 1 robin robin 55 Sep 11 14:23 files
robin@robin-VirtualBox:~/Music$
```

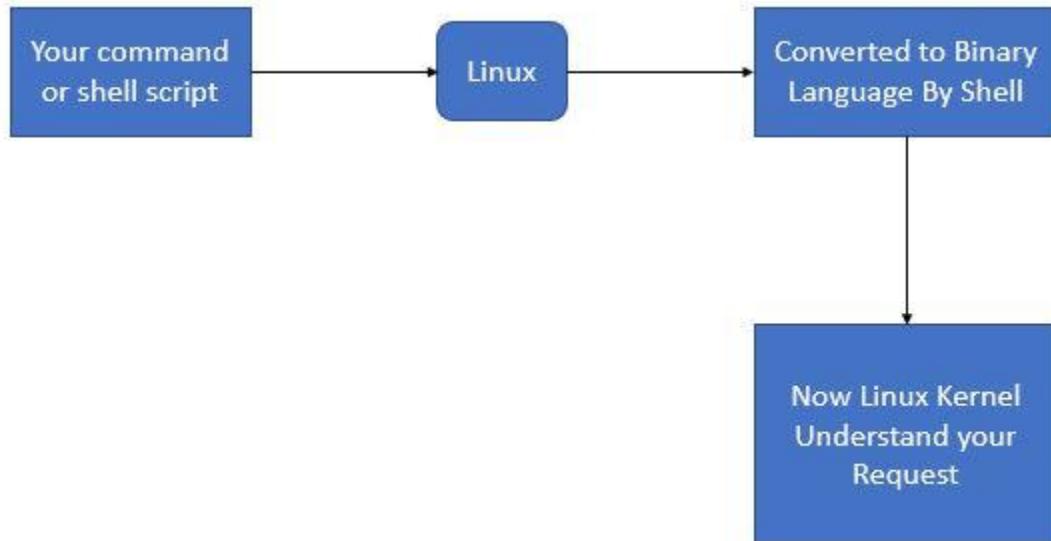
EXPERIMENT NO: 13

FAMILIARISATION TO LINUX SHELL AND SHELL SCRIPTING

Shell

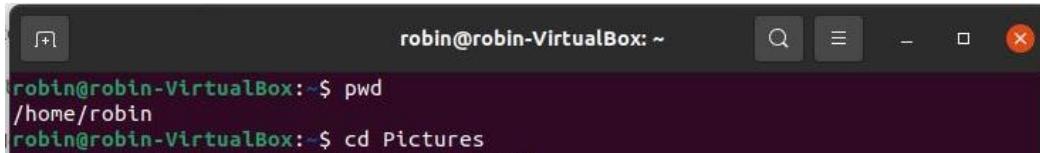
- A shell is special user program which provide an interface to user to use operating system services.
- Shell accept human readable commands from user and convert them into something which kernel can understand.
- shell gets started when the user logs in or start the terminal.
- Shell is broadly classified into two categories
 - * Command Line Shell
 - * Graphical shell

This is what Shell Does for US



- The shell in the linux operating takes input in the form of commands process them and gives an output .

- When you are in the terminal the shell issues an command prompt.
- which is in usually \$ where in you can type the input which is then executed by hitting Enter key.



```
robin@robin-VirtualBox:~$ pwd
/home/robin
robin@robin-VirtualBox:~$ cd Pictures
```

Bash

- BASH (Bourne Again SHell) –
- It is most widely used shell in Linux systems.
- It is used as default login shell in Linux systems.
- It can also be installed on Windows OS.

Shell Scripting

- Shell script is a series of command(s) stored in a plain text file.
- These files are called Shell Scripts or Shell Programs.
- Each shell script is saved with .sh file extension .
- eg. myscript.sh

Creating a shell script

- Creating a file using any text editor
- Start the script with #!/bin/sh
- Write some code
- Save the script file as **filename.sh**
- For executing the script type **bash filename.sh**

Variables in Shell

- Variables store data in the form of character and number.
- In Linux (Shell), there are two types of variable:
 - **System variables** - Created and maintained by Linux itself. This type of variable defined in CAPITAL LETTERS.

- **User defined variables (UDV)** - Created and maintained by user. This type of variable defined in lower letters

The read Statement

- Use to get input (data from user) from keyboard and store (data) to variable.
- Syntax:
 - **read** variable1, variable2,...variableN

Example

```
echo "Your first name please:"  
read fname  
echo "Hello $fname, Lets be friend!"
```

Conditional Statements

- **if** condition
 - used for making decisions in shell script,
 - If the condition is true
 - then command1 is executed.

Syntax:

```
if condition  
then  
command1 if condition is true or if exit status of condition is 0  
fi
```

The screenshot shows a terminal window with the title bar "sample.sh" and the path "~/Pictures". The file contains the following shell script:

```

1 #!/bin/sh
2 echo "Enter the age"
3 read age
4 if [ $age -ge 18 ]
5 then
6     echo "Eligible"
7 fi

```

The screenshot shows a terminal window with the title bar "robin@robin-VirtualBox: ~/Pictures". The command "bash sample.sh" was run, and the output is:

```

robin@robin-VirtualBox:~$ cd Pictures
robin@robin-VirtualBox:~/Pictures$ bash sample.sh
Enter the age
21
Eligible
robin@robin-VirtualBox:~/Pictures$ 

```

if...else...fi

- If given condition is true
 - then command1 is executed
 - otherwise command2 is executed.
- Syntax:

if condition

then

condition is zero (true - 0)

execute all commands up to else statement

else

if condition is not true then

execute all commands up to fi

fi

The screenshot shows a terminal window with the following content:

```

1 #!/bin/sh
2 echo "Enter the age"
3 read age
4 if [ $age -ge 18 ]
5 then
6     echo "Eligible"
7 else
8     echo "Not eligible"
9 fi

```

robin@robin-VirtualBox:~/Pictures\$ cd Pictures
robin@robin-VirtualBox:~/Pictures\$ bash if_else.sh
Enter the age
12
Not eligible
robin@robin-VirtualBox:~/Pictures\$

Loops in Shell Scripts

- Bash supports:
 - for loop
 - while loop

Note that in each and every loop,

- First, the variable used in loop condition must be initialized, then execution of the loop begins.
- A test (condition) is made at the beginning of each iteration.
- The body of loop ends with a statement that modifies the value of the test (condition) variable.

for Loop

- Syntax:

for ((*expr1*; *expr2*; *expr3*))

do

repeat all statements between do and done until expr2 is TRUE

done

The screenshot shows a code editor window with two tabs. The active tab is 'for.sh' located in the '/Pictures' directory. The script contains the following code:

```
1 #!/bash/sh
2 echo "Enter the limit :"
3 read n
4 echo -----
5 for (( i=1; i <= n ; i++ ))
6 do
7   echo "$i"
8 done
```

The status bar at the bottom indicates 'sh' as the file type, 'Tab Width: 8', 'Ln 6, Col 4', and 'INS' for insert mode.

The screenshot shows a terminal window titled 'robin@robin-VirtualBox: ~/Pictures'. The user runs the command 'bash for.sh'. The terminal outputs the following:

```
robin@robin-VirtualBox:~/Pictures$ bash for.sh
Enter the limit :
10
-----
1
2
3
4
5
6
7
8
9
10
```

while Loop

- Syntax:

while [condition]

do

command1

command2

..

....

done

The screenshot shows a terminal window with the following content:

```

1#!/bin/sh
2a=1
3while [ $a -le 10 ]
4do
5
6echo "$a"
7a=$((a+1))
8
9done

```

Below the code, the terminal shows the command: `sh -c "cat while.sh" > while.sh`. The status bar at the bottom right indicates "Ln 3, Col 15".

The screenshot shows a terminal window with the following content:

```

robin@robin-VirtualBox:~/Pictures$ bash while.sh
1
2
3
4
5
6
7
8
9
10
robin@robin-VirtualBox:~/Pictures$ █

```

The terminal prompt is `robin@robin-VirtualBox:~/Pictures$`.

Functions

- Function is series of instruction/commands.
- Function performs particular activity in shell.
- Code in a function is only executed when a function is ‘called’

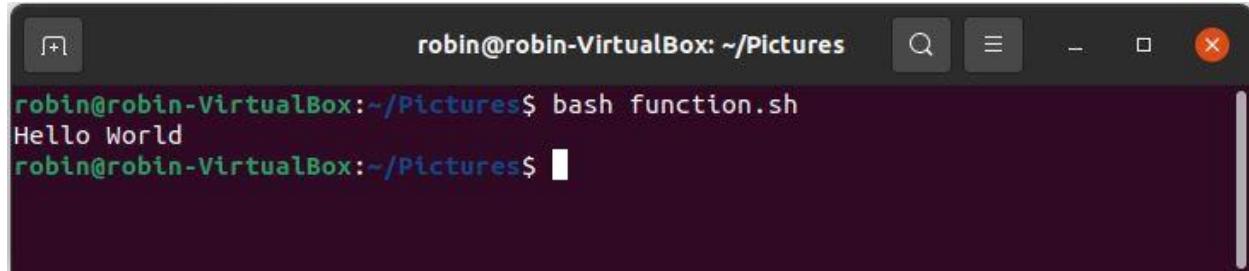
Syntax:

function-name ()

{

Function body

}



A terminal window titled "robin@robin-VirtualBox: ~/Pictures". The command "bash function.sh" is run, followed by the output "Hello World".

```
robin@robin-VirtualBox:~/Pictures$ bash function.sh
Hello World
robin@robin-VirtualBox:~/Pictures$
```



A code editor window titled "function.sh" located in the "~/Pictures" directory. The file contains the following script:

```
1#!/bash/sh
2 hello () {
3 echo "Hello World"
4 }
5 hello
```

The status bar at the bottom shows "sh" as the shell, "Tab Width: 8", "Ln 5, Col 7", and "INS" for insert mode.

Function(using argument) - Example 2



A terminal window titled "robin@robin-VirtualBox: ~/Pictures". The command "bash function.sh" is run, followed by the prompt "Enter the numbers to be add:". The user then enters "10" and "5", and the script outputs "Sum: 15".

```
robin@robin-VirtualBox:~/Pictures$ bash function.sh
Enter the numbers to be add:
10
5
Sum: 15
robin@robin-VirtualBox:~/Pictures$
```



```

1 #!/bash/sh
2 Addition () {
3     s=$((a+b))
4     echo "Sum: $s"
5 }
6 echo "Enter the numbers to be add:"
7 read a
8 read b
9 Addition $a $b

```

sh ▾ Tab Width: 8 ▾ Ln 5. Col 2 ▾ INS

Command Line Arguments



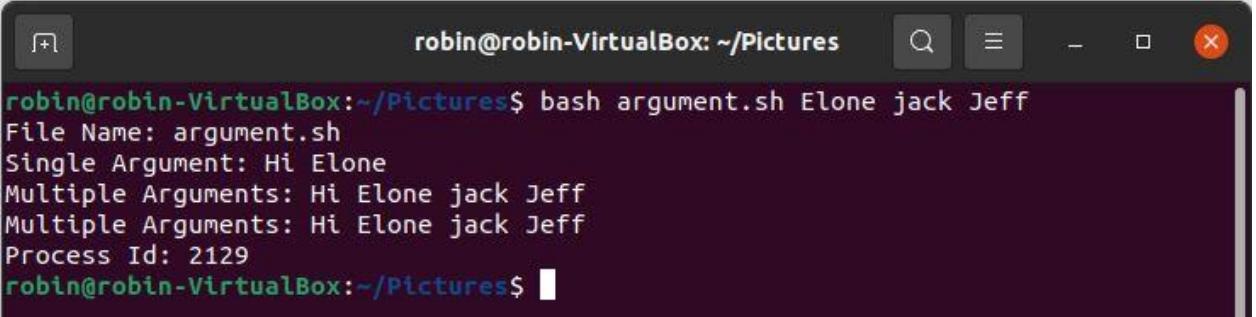
```

1 #!/bin/bash
2 echo "File Name: $0"
3 echo "Single Argument: Hi $1"
4 echo "Multiple Arguments: Hi $*"
5 echo "Multiple Arguments: Hi $@"
6 echo "Process Id: $$"

```

sh ▾ Tab Width: 8 ▾ Ln 6, Col 22 ▾ INS

Command	Explanation
\$0	Represent the command or script.
\$1	Represent argument 1.
\$#	Represents the total number of argument
\$*	Represents all arguments.
\$\$	Represents the PID of a running script.



```
robin@robin-VirtualBox:~/Pictures$ bash argument.sh Elone jack Jeff
File Name: argument.sh
Single Argument: Hi Elone
Multiple Arguments: Hi Elone jack Jeff
Multiple Arguments: Hi Elone jack Jeff
Process Id: 2129
robin@robin-VirtualBox:~/Pictures$
```

Mathematical Operators

Mathematical Operator in Shell Script	Meaning	Normal Arithmetical/Mathematical Statements	But in Shell	
			For test statement with if command	For [expr] statement with if command
-eq	is equal to	5 == 6	if test 5 -eq 6	if [5 -eq 6]
-ne	is not equal to	5 != 6	if test 5 -ne 6	if [5 -ne 6]
-lt	is less than	5 < 6	if test 5 -lt 6	if [5 -lt 6]
-le	is less than or equal to	5 <= 6	if test 5 -le 6	if [5 -le 6]
-gt	is greater than	5 > 6	if test 5 -gt 6	if [5 -gt 6]
-ge	is greater than or equal to	5 >= 6	if test 5 -ge 6	if [5 -ge 6]

String Operators

Operator	Meaning
string1 = string2	string1 is equal to string2
string1 != string2	string1 is NOT equal to string2
string1	string1 is NOT NULL or not defined
-n string1	string1 is NOT NULL and does exist
-z string1	string1 is NULL and does exist

Logical Operators

Operator	Meaning
string1 = string2	string1 is equal to string2
string1 != string2	string1 is NOT equal to string2
-n string1	string1 is NOT NULL or not defined
-z string1	string1 is NOT NULL and does exist
-e string1	string1 is NULL and does not exist

EXPERIMENT NO:14

LINUX SHELL SCRIPITING PROBLEMS

1. Write a shell script to get current date, time, username and current working?

Ans :

```
#!/bin/bash
echo "hello,$LOGNAME"
echo "Current date is `date` "
echo "User `who am i` "
echo "Current directory `pwd` "
```

output:

```
robin@robin-VirtualBox:~/Pictures$ bash 1.sh
hello,robin
Current date is Thursday 12 August 2021 06:42:38 PM IST
User is
Current directory /home/robin/Pictures
robin@robin-VirtualBox:~/Pictures$
```

2. How to print the login names of all users on a system?

Ans :

```
cat /etc/passwd
```

Output:

```
robin@robin-VirtualBox:~/Pictures$ bash 2.sh
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuid:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi.v.112.120.Avahi mDNS daemon ->/run/avahi-daemon:/usr/sbin/nologin
```

3. How can we pass arguments to a script in Linux? And how to access these arguments from within the script?

Ans :

```
echo "first number : $1"
echo "second number : $2"
echo "third number : $3"
```

Output:

```
robin@robin-VirtualBox:~/Pictures$ bash 3.sh 20 50 40
first parameter is 20
second parameter is 50
third parameter is 40
robin@robin-VirtualBox:~/Pictures$
```

4. How to set an array in Linux?

Ans :

```
array=(hello i am robin)
echo ${array[@]}
echo ${array[*]}
echo ${array[@]:0}
echo ${array[*]:0}
echo ${array[0]}
echo ${array[1]}
echo ${array[2]}
echo ${array[3]}
```

output :

```
robin@robin-VirtualBox:~/Pictures$ bash 4.sh
hello i am robin
hello i am robin
hello i am robin
hello i am robin
hello
i
am
robin
robin@robin-VirtualBox:~/Pictures$ █
```

5. How to check if a directory exists?

Ans :

```
if [ -d "/home/user/shellscriptingpgms" ]
then
    echo "/home/user/shellscriptingpgms exists."
else
    echo "Error: Directory /home/user/shellscriptingpgms does not exists."
fi
```

Output :

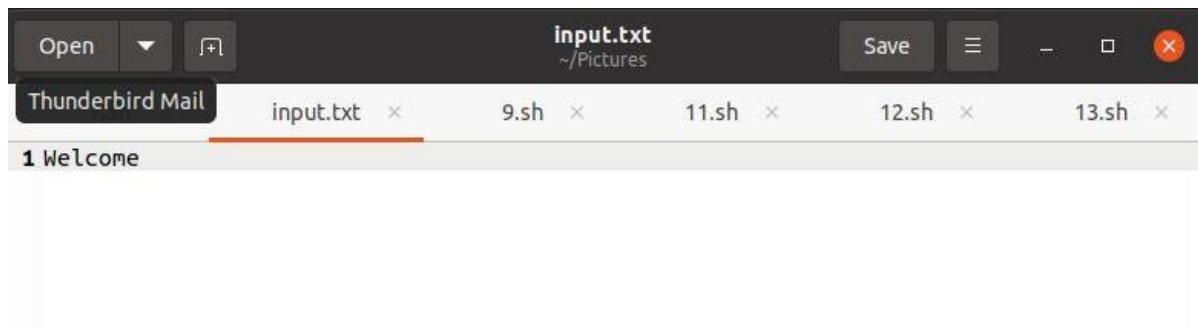
```
robin@robin-VirtualBox:~/Pictures$ bash 5.sh
Directory/home/robin/Pictures exits
robin@robin-VirtualBox:~/Pictures$
```

6. What is the difference between \$* and \$@?

Ans:

\$@ treats each quoted arguments as separate arguments but \$* considers the entire set of positional parameters as a single string.

7. Use the sed command to replace the content of the file?

Ans:

```
robin@robin-VirtualBox:~/Pictures$ sed -i 's/Welcome/This is my world/g' input.txt
robin@robin-VirtualBox:~/Pictures$
```

```
Open    input.txt ~Pictures Save - X
*7.sh x   input.txt x   9.sh x   11.sh x   12.sh x   13.sh x
1 This is my world
```

8. Write a script to compare numbers?

Ans :

```
var1=100
var2=200
if [ $var2 -gt $var1 ]
then
echo "$var2 is greater than $var1"
fi
#second comparison
echo "_____"
if [ $var1 -gt 1000 ]
then
echo "$var1 is greater than 1000"
else
echo
echo "$var1 is less than 1000"
fi
```

Output:

```
robin@robin-VirtualBox:~/Pictures$ bash 8.sh
20 is greater than 10
10 is less than 30
robin@robin-VirtualBox:~/Pictures$
```

9. Write a shell script to check to see if the file “file_path” exists. If it does exist, display “file_path passwords are enabled.” Next, check to see if you can write to the file. If you can, display “You have permissions to edit “file_path.”” If you cannot, display “You do NOT have permissions to edit“file_path””?

Ans :

```
FILE="/home/user/shells scripting pgms/pgm1"
if [ -e "$FILE" ]
then
echo "$FILE passwords are enabled"
fi
if [ -x "$FILE" ]
then
echo "You have permission to execute $FILE"
else
echo "You have no permission to execute $FILE"
fi
```

Output :

```
robin@robin-VirtualBox:~/Pictures$ bash 9.sh
/home/robin/Pictures/add.c passwords are enabled
You have no permission to execute /home/robin/Pictures/add.c
robin@robin-VirtualBox:~/Pictures$
```

10. How to print all array indexes?

Ans :

```
arr=(10 20 30 40 50)
echo "index : value"
for((i=0;i<${#arr[@]};i++))
```

```

do
echo "$i : ${arr[$i]}"
done

```

output :

```

robin@robin-VirtualBox:~/Pictures$ bash 10.sh
index : value
0 : 10
1 : 20
2 : 30
3 : 40
4 : 50
robin@robin-VirtualBox:~/Pictures$ █

```

11 . Write a shell script to display the last updated file or the newest file in a directory?

Ans :

```
ls -lrt | grep ^- | awk '{print $NF}'
```

output:

```

robin@robin-VirtualBox:~/Pictures$ bash 11.sh
input.txt
robin@robin-VirtualBox:~/Pictures$ █

```

12. Write a shell script that adds an extension “.new” to all the files in directory.

Ans :

```

dir=$1
for file in '$1/*'
do
mv $file $file.new
done

```

13. Write a shell script to print a number in reverse order. It should support

- the following requirements.
- The script should accept the input from the command line.

- c. If you don't input any data, then display an error message to execute the script correctly

Ans :

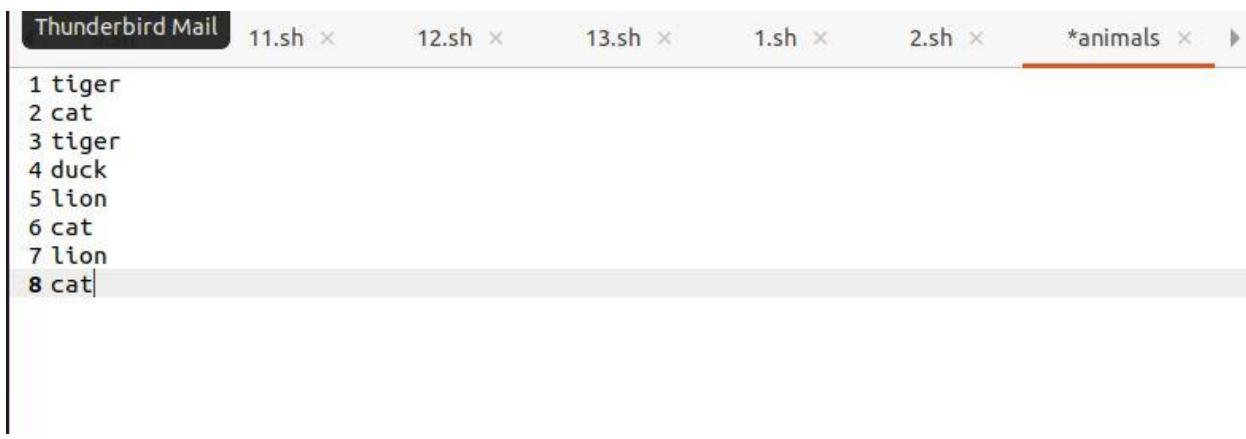
```
echo "Enter a number to be reversed :"  
read n  
if [ $n == 0 ]  
then  
echo "Please provide the correct input in the below format."  
echo "Usage: $0 number"  
echo " This script will reverse the given number."  
echo " For eg. $0 1234, will print 4321"  
exit 1  
  
else  
rev=0  
sd=0  
while [ $n -gt 0 ]  
do  
sd=`expr $n % 10`  
rev=`expr $rev \* 10 + $sd`  
n=`expr $n / 10`  
done  
echo "Reverse number is $rev"  
fi
```

Output:

```
robin@robin-VirtualBox:~/Pictures$ bash 13.sh
Enter a number to be reversed :
456
Reverse number is 654
robin@robin-VirtualBox:~/Pictures$
```

14. Write a shell script delete a file which has special characters in its file name

15. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words

Ans:**Output:**

```
robin@robin-VirtualBox:~/Pictures$ sed -e 's/ /\n/g' animals | grep -v '^$' | sort | uniq -c | sort -n
   1 duck
   2 lion
   2 tiger
   3 cat
robin@robin-VirtualBox:~/Pictures$
```

16. Write a script to print the first 10 elements of Fibonacci series.

Ans :

N=10

a=0

b=1

echo "The Fibonacci series is : "

```

for (( i=1; i<=N; i++ ))
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done

```

Output:

```

robin@robin-VirtualBox:~/Pictures$ bash 16.sh
The Fibonacci series is :
0 1 1 2 3 5 8 13 21 34 robin@robin-VirtualBox:~/Pictures$ █

```

17. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

18. Write a shell script to validate password strength. Here are a few assumptions for the password string.

Length – minimum of 8 characters. Contain both alphabet and number.

Include both the small and capital case letters.

If the password doesn’t comply with any of the above conditions, then the script should report it as a <Weak Password>.

Ans :

```

echo "Enter your password"
read password

```

```
len="${#password}"\n\nif test $len -ge 8 ; then\n    echo "$password" | grep -q [0-9]\n    if test $? -eq 0 ; then\n        echo "$password" | grep -q [A-Z]\n        if test $? -eq 0 ; then\n            echo "$password" | grep -q [a-z]\n            if test $? -eq 0 ; then\n                echo "Strong Password"\n            else\n                echo "Weak Password -> Should include a lower case letter."\n            fi\n        else\n            echo "Weak Password -> Should include a capital case letter."\n        fi\n    else\n        echo "Weak Password -> Should use numbers in your password."\n    fi\nelse\n    echo "Weak Password -> Password length should have at least 8 characters."\nfi
```

Output:

```
robin@robin-VirtualBox:~/Pictures$ bash 18.sh\nEnter your password\nAlexa4356\nStrong Password\nrobin@robin-VirtualBox:~/Pictures$ █
```

EXPERIMENT NO: 15 INSTALLATION OF LAMP

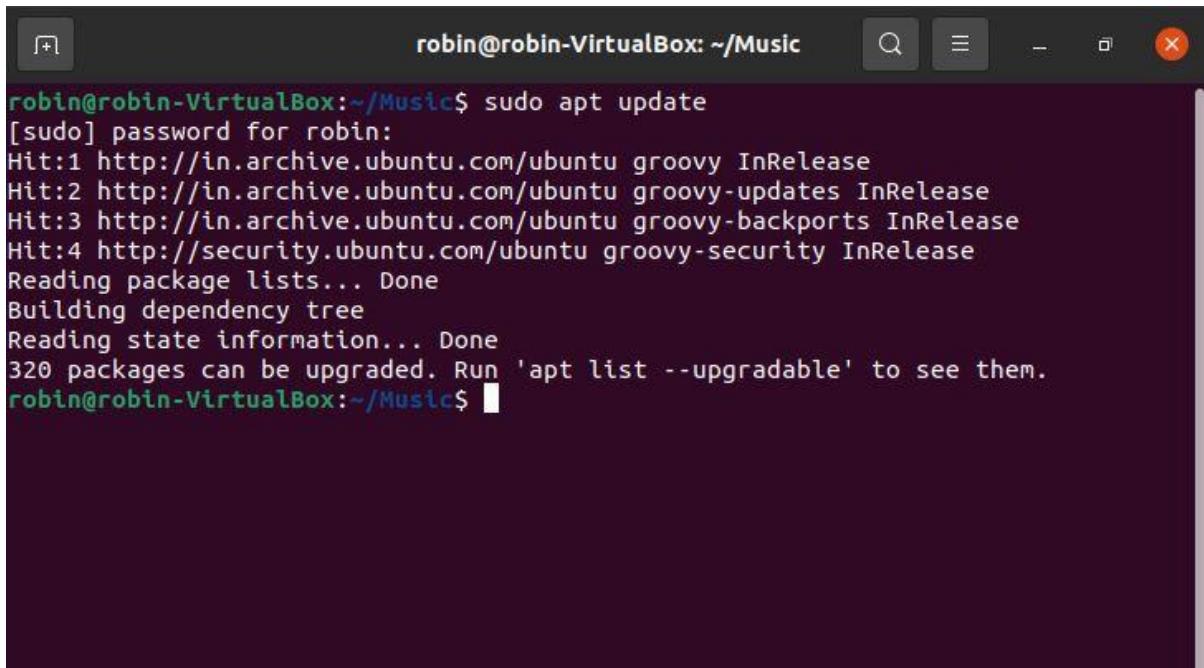
Introduction

A “LAMP” stack is a group of open-source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the Linux operating system, with the Apache web server. The site data is stored in a MySQL database, and dynamic content is processed by PHP

Installing Apache and Updating the Firewall

Update cache using

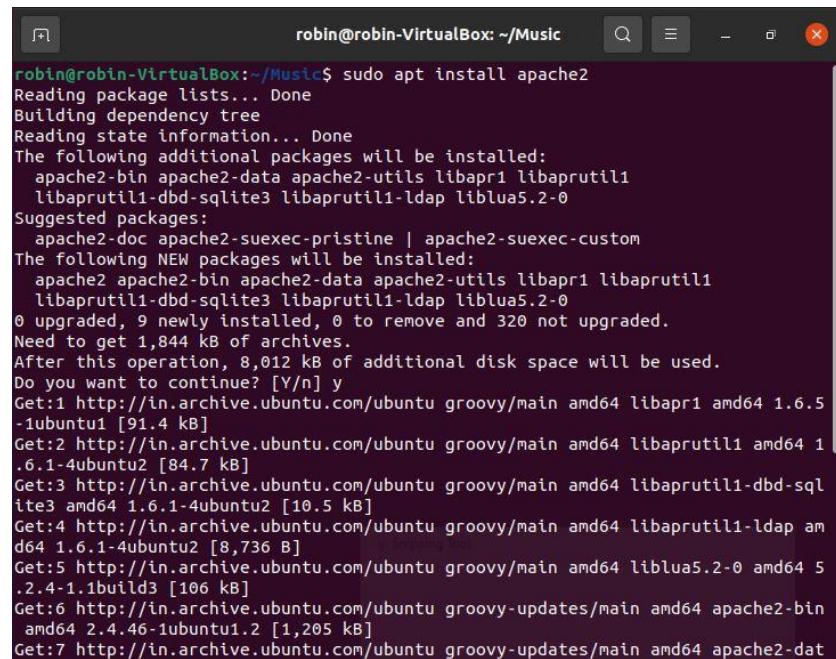
\$ sudo apt update



The screenshot shows a terminal window titled "robin@robin-VirtualBox: ~/Music". The command "sudo apt update" is run, followed by a password entry prompt "[sudo] password for robin:". The output shows the process of updating package lists from various repositories, including hits for InRelease, InRelease-updates, InRelease-backports, and InRelease-security files. It also shows the reading of package lists and dependency trees, and concludes with a message stating 320 packages can be upgraded. The terminal ends with the prompt "robin@robin-VirtualBox:~/Music\$".

Once the cache has been updated, you can install Apache with:

- **\$ sudo apt install apache2**



```
robin@robin-VirtualBox:~/Music$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 320 not upgraded.
Need to get 1,844 kB of archives.
After this operation, 8,012 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libapr1 amd64 1.6.5-1ubuntu1 [91.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libaprutil1 amd64 1.6.1-4ubuntu2 [84.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-4ubuntu2 [10.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libaprutil1-ldap amd64 1.6.1-4ubuntu2 [8,736 B]
Get:5 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 liblua5.2-0 amd64 5.2.4-1.1build3 [106 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 apache2-bin amd64 2.4.46-1ubuntu1.2 [1,205 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 apache2-data
```

Adjust the Firewall to Allow Web Traffic

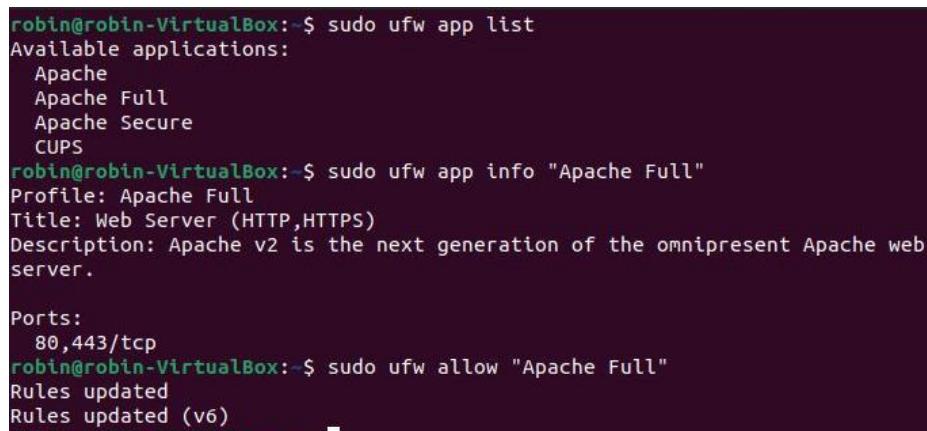
\$ sudo ufw app list

If you look at the Apache Full profile details, you'll see that it enables traffic to ports 80 and 443:

\$ sudo ufw app info "Apache Full"

To allow incoming HTTP and HTTPS traffic for this server, run:

- **sudo ufw allow "Apache Full"**



```
robin@robin-VirtualBox:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
robin@robin-VirtualBox:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web server.

Ports:
  80,443/tcp
robin@robin-VirtualBox:~$ sudo ufw allow "Apache Full"
Rules updated
Rules updated (v6)
robin@robin-VirtualBox:~$
```

Installing MySQL

Now the web server is up and running and its time to install MySQL.

Again, use apt to acquire and install this software:

- **\$ sudo apt install mysql-server**

```
robin@robin-VirtualBox:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgifast-perl libcgipm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  libprotobuf-lite23 mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-server-8.0
  mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgifast-perl libcgipm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  libprotobuf-lite23 mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 upgraded, 17 newly installed, 0 to remove and 320 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 255 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.25-0ubuntu0.20.10.1 [4,202 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 mysql-client-8.0 amd64 8.0.25-0ubuntu0.20.10.1 [22.0 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libaio1 amd64 0.3.1-12-8 [7,488 B]
Get:4 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 libevent-core-2.1-7
```

When the installation completes, run a script that will remove some dangerous defaults and lockdown the access to your database

\$ sudo mysql_secure_installation

```
robin@robin-VirtualBox:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW   Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

When you're finished, test if you're able to log in to the MySQL console by typing:

- **\$ sudo mysql**

To exit the MySQL console, type:

- **\$ exit**



```
robin@robin-VirtualBox:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.25-0ubuntu0.20.10.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

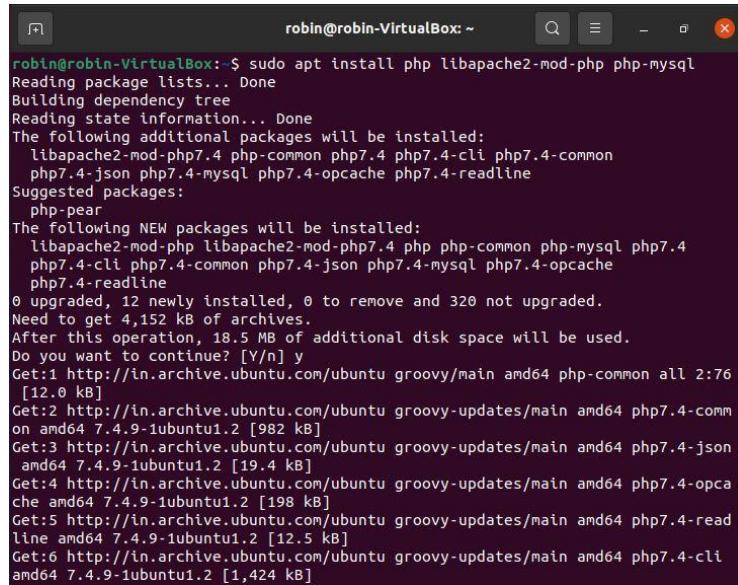
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> /c
      -> tata
      -> /c
/c
^C
mysql> exit
Bye
robin@robin-VirtualBox:~$
```

Installing PHP

Now install PHP and libapache2-mod-php to integrate PHP to Apache and the php-mysql package to allow PHP to connect to MySQL.

\$ sudo apt install php libapache2-mod-php php-mysql



```
robin@robin-VirtualBox:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common
  php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.4 php php-common php-mysql php7.4
  php7.4-cli php7.4-common php7.4-json php7.4-mysql php7.4-opcache
  php7.4-readline
0 upgraded, 12 newly installed, 0 to remove and 320 not upgraded.
Need to get 4,152 kB of archives.
After this operation, 18.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 php-common all 2:7.6.0-12.0 [12.0 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 php7.4-common 7.4.9-1ubuntu1.2 [982 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 php7.4-json 7.4.9-1ubuntu1.2 [19.4 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 php7.4-opcache 7.4.9-1ubuntu1.2 [198 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 php7.4-readline 7.4.9-1ubuntu1.2 [12.5 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 php7.4-cli 7.4.9-1ubuntu1.2 [1,424 kB]
```

This should install PHP without any problems. We'll test this

EXPERIMENT NO:16

INSTALLATION OF LARAVEL

LARAVEL

- Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.
- Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.
- If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot of time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

ADVANTAGES OF LARAVEL

- Laravel offers you the following advantages, when you are designing a web application based on it –
- The web application becomes more scalable, owing to the Laravel framework.
- Considerable time is saved in designing the web application, since Laravel reuses the components from other frameworks in developing web applications.
- It includes namespaces and interfaces, thus help to organize and manage resources.

COMPOSER

- Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

INSTALLATION OF LARAVEL

- Step 1 – Visit the following URL and download composer to install it on your system.

<https://getcomposer.org/download/>

Remember to set the path of the composer to the php.exe file in the php/xampp/C:

- Step 2 – After the Composer is installed, check the installation by typing the Composer command in the command prompt as shown in the following screenshot.

```

C:\ Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\X541U>composer

Composer version 2.1.8 2021-09-15 13:55:14

Usage:
  command [options] [arguments]

Options:
  -h, --help           Display this help message
  -q, --quiet          Do not output any message
  -V, --version         Display this application version
  --ansi               Force ANSI output
  --no-ansi             Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins          Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache            Prevent use of the cache
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about      Shows a short information about Composer.
  archive    Creates an archive of this composer package.
  browse     Opens the package's repository URL or homepage in your browser.
  cc         Clears composer's internal package cache.
  check-platform-reqs Check that platform requirements are satisfied.
  clear-cache Clears composer's internal package cache.
  clearcache Clears composer's internal package cache.
  config     Sets config options.
  create-project Creates new project from a package into given directory.
  depends    Shows which packages cause the given package to be installed.
  diagnose   Diagnoses the system to identify common errors.
  dump-autoload Dumps the autoloader.
  dumpautoload Dumps the autoloader.
  exec       Executes a vendored binary/script.
  fund       Discover how to help fund the maintenance of your dependencies.

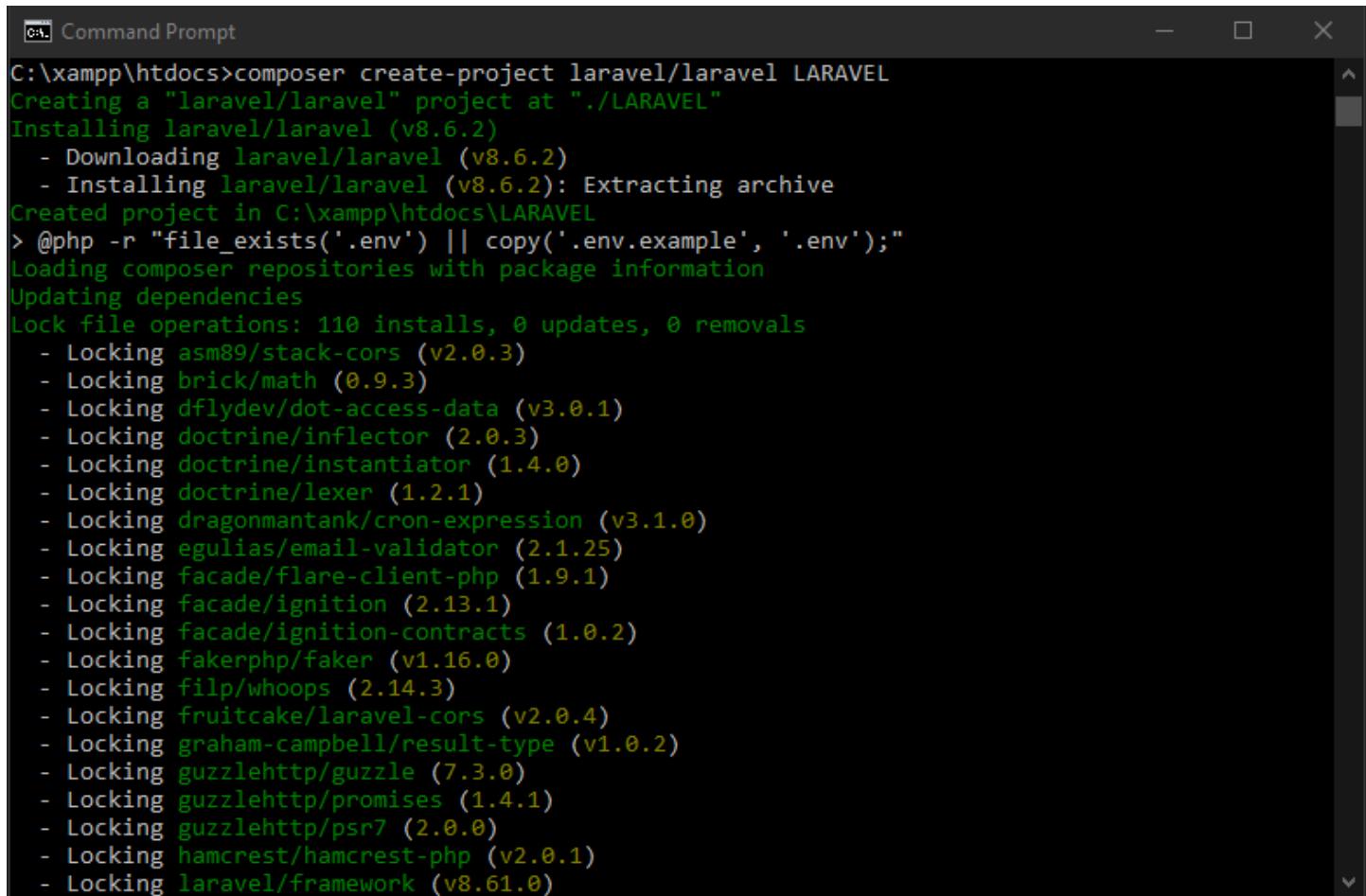
```

- Step 3 – Create a new directory anywhere in your system for your new Laravel project. After that, move to the path where you have created the new directory and type the following command there to install Laravel.

composer create-project --prefer-dist laravel/laravel folder_name

But the latest version(currently 8.x) just requires you to type

composer create-project laravel/laravel folder_name

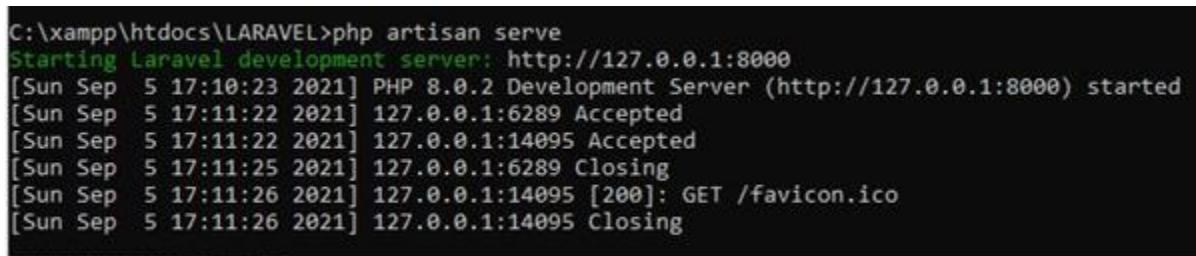


```
C:\xampp\htdocs>composer create-project laravel/laravel LARAVEL
Creating a "laravel/laravel" project at "./LARAVEL"
Installing laravel/laravel (v8.6.2)
- Downloading laravel/laravel (v8.6.2)
- Installing laravel/laravel (v8.6.2): Extracting archive
Created project in C:\xampp\htdocs\LARAVEL
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.0.3)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.13.1)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.16.0)
- Locking filp/whoops (2.14.3)
- Locking fruitcake/laravel-cors (v2.0.4)
- Locking graham-campbell/result-type (v1.0.2)
- Locking guzzlehttp/guzzle (7.3.0)
- Locking guzzlehttp/promises (1.4.1)
- Locking guzzlehttp/psr7 (2.0.0)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v8.61.0)
```

- Step 4 – The above command will install Laravel in the current directory. Start the Laravel service by executing the following command.

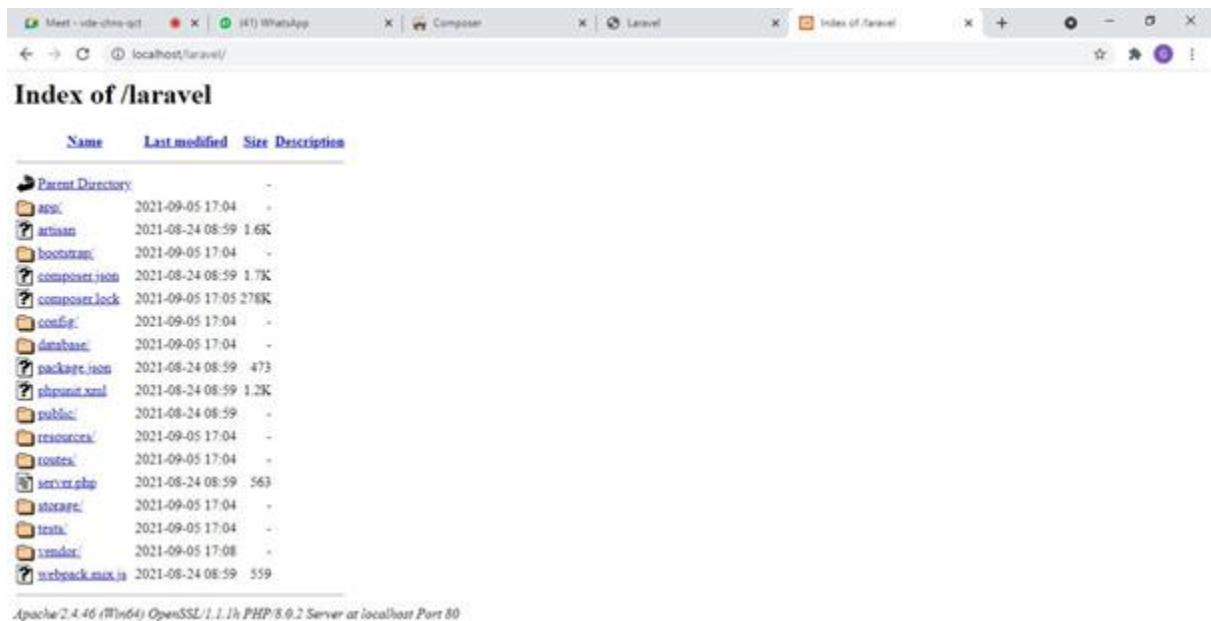
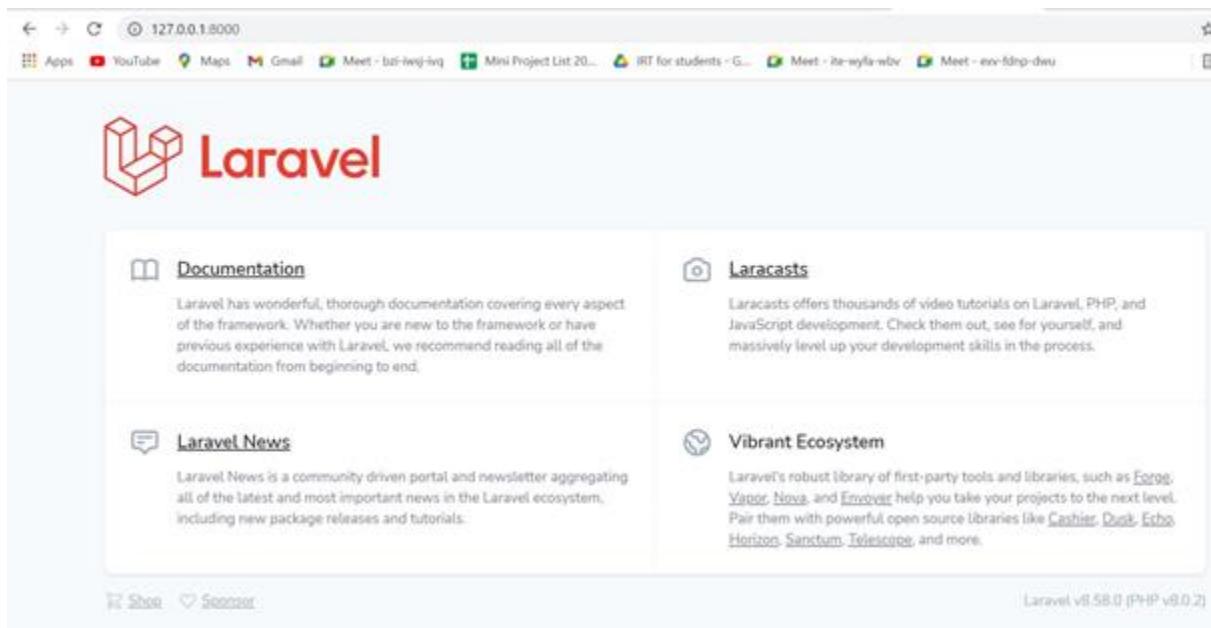
php artisan serve

- Step 5 – After executing the above command, you will see a screen as shown below –



```
C:\xampp\htdocs\LARAVEL>php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sun Sep 5 17:10:23 2021] PHP 8.0.2 Development Server (http://127.0.0.1:8000) started
[Sun Sep 5 17:11:22 2021] 127.0.0.1:6289 Accepted
[Sun Sep 5 17:11:22 2021] 127.0.0.1:14095 Accepted
[Sun Sep 5 17:11:25 2021] 127.0.0.1:6289 Closing
[Sun Sep 5 17:11:26 2021] 127.0.0.1:14095 [200]: GET /favicon.ico
[Sun Sep 5 17:11:26 2021] 127.0.0.1:14095 Closing
```

- Step 6 – Copy the URL underlined in gray in the above screenshot and open that URL in the browser. If you see the following screen, it implies Laravel has been installed successfully.



- Step 7 – The following screen indicates the laravel framework has been successfully installed in your device.

EXPERIMENT NO: 17

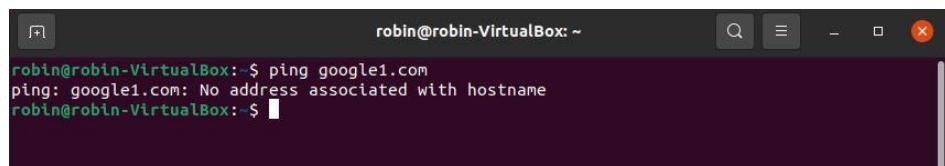
FAMILIARIZATION TO COMMAND LINE TOOL FOR NETWORKING

NETWORK COMMANDS

- The operating system consists of various built-in, command-line networking utilities that are used for network troubleshooting.
- **1. Ping**
- **2. Hostname**
- **3.netstat**
- **4. If Config**
- **5. Nslookup**
- **6. Traceroute**
- **7.Route**

PING

- PACKET INTERNET GROPER
- Most widely used utility tool to troubleshoot
- It sends packets of information to the user-defined source. If the packets are received, the destination device sends packets back.
- Ping can be used for two purposes
 - Network connection can be established
 - Speed of the connection
- Ping is used to testing a network host capacity to interact with another host. Just enter the command Ping, followed by the target host's name or IP address.



A screenshot of a terminal window titled "robin@robin-VirtualBox: ~". The window contains the following text:
robin@robin-VirtualBox: \$ ping google1.com
ping: google1.com: No address associated with hostname
robin@robin-VirtualBox: \$ █

```

robin@robin-VirtualBox:~$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=63 time=1.03 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=63 time=2.92 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=63 time=2.07 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=63 time=2.30 ms
64 bytes from 192.168.56.1: icmp_seq=5 ttl=63 time=1.93 ms
^C
--- 192.168.56.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.026/2.050/2.924/0.614 ms
robin@robin-VirtualBox:~$ ping google.com
PING google.com (142.250.67.78) 56(84) bytes of data.
64 bytes from maa05s13-in-f14.1e100.net (142.250.67.78): icmp_seq=1 ttl=112 time=67.9 ms
64 bytes from maa05s13-in-f14.1e100.net (142.250.67.78): icmp_seq=2 ttl=112 time=68.8 ms
64 bytes from maa05s13-in-f14.1e100.net (142.250.67.78): icmp_seq=3 ttl=112 time=67.4 ms
64 bytes from maa05s13-in-f14.1e100.net (142.250.67.78): icmp_seq=4 ttl=112 time=94.8 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 67.445/74.707/94.779/11.597 ms
robin@robin-VirtualBox:~$ 
```

HOSTNAME

- Hostname :displays the machine hostname
- Hostname –f :displays the fully qualified host and domain name
- Hostname –I :displays the ip address for the current machine

```

robin@robin-VirtualBox:~$ ping google1.com
ping: google1.com: No address associated with hostname
robin@robin-VirtualBox:~$ hostnamectl
      Static hostname: robin-VirtualBox
                  Icon name: computer-vm
                    Chassis: vm
           Machine ID: 718669f652b24f6c81b1786485e3dca7
              Boot ID: f46275d85712412eab7047c2a2a45df2
        Virtualization: oracle
      Operating System: Ubuntu 20.10
                 Kernel: Linux 5.8.0-25-generic
             Architecture: x86-64
robin@robin-VirtualBox:~$ hostname
robin-VirtualBox
robin@robin-VirtualBox:~$ hostname -f
robin-VirtualBox
robin@robin-VirtualBox:~$ hostname -i
127.0.1.1
robin@robin-VirtualBox:~$ 
```

Netstat

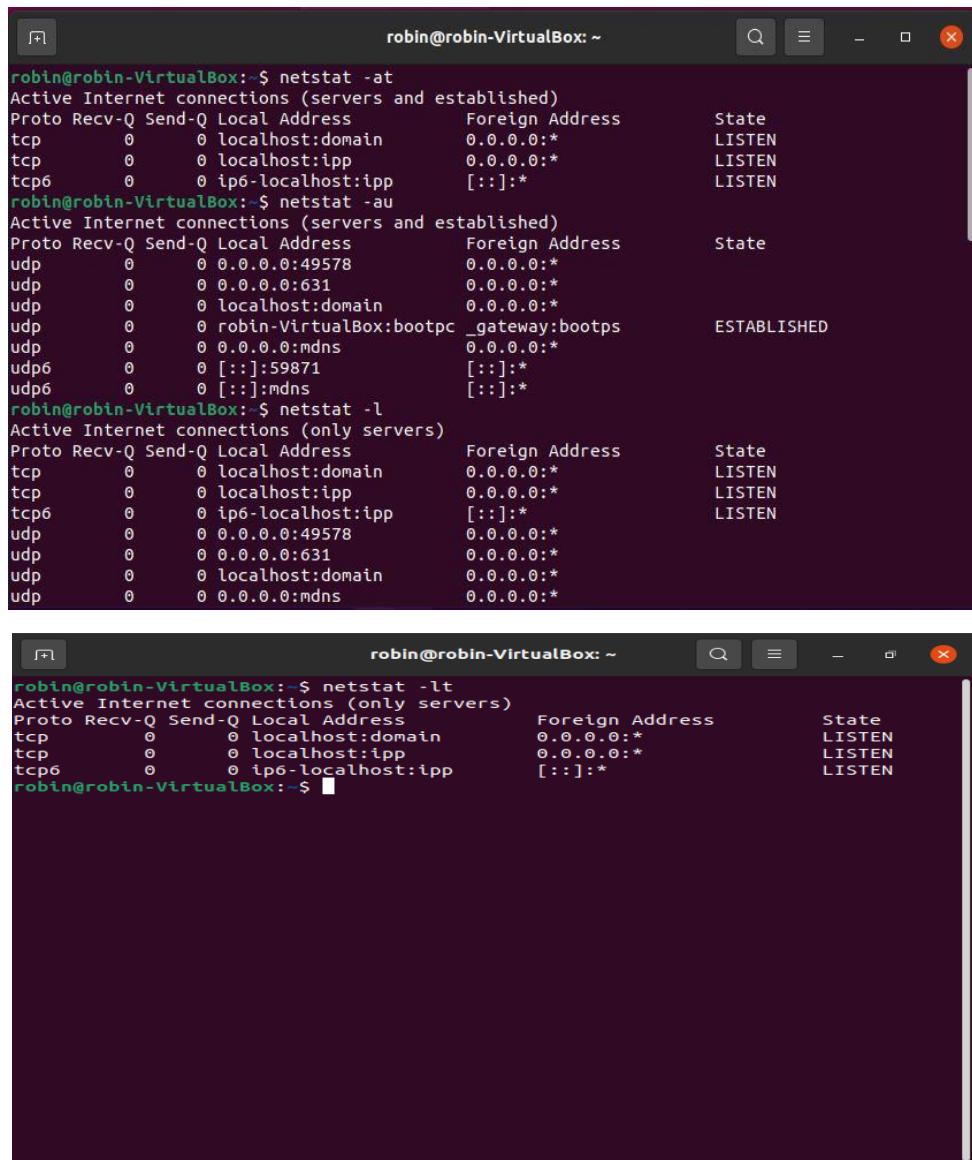
- Netstat command displays various network related information such as network connections, routing tables, interface statistics, multicast memberships etc.
- **netstat -a** : To show both listening and non-listening sockets
- **netstat -at** : To list all tcp ports.
- **netstat -au** : To list all udp ports.
- **netstat -l** : To list only the listening ports.
- **netstat -lt** : To list only the listening tcp ports
- **netstat -lu** : To list only the listening udp ports.
- **netstat -lx** : To list only the listening UNIX ports.
- **netstat -s** : To list the statistics for all ports

```

robin@robin-VirtualBox:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
udp      0      0 robin-VirtualBox:bootpc _gateway:bootps      ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State         I-Node Path
unix    2      [ ]     DGRAM          29455  /run/user/1000/systemd/notify
unix    3      [ ]     DGRAM          15700  /run/systemd/notify
unix    2      [ ]     DGRAM          15714  /run/systemd/journal/syslog
unix   17      [ ]     DGRAM          15723  /run/systemd/journal/dev-log
unix    8      [ ]     DGRAM          15727  /run/systemd/journal/socket
unix    3      [ ]     STREAM   CONNECTED    31132  /run/systemd/journal/stdout
unix    3      [ ]     STREAM   CONNECTED    35661  /run/systemd/journal
unix    3      [ ]     STREAM   CONNECTED    33734  /run/user/1000/bus
unix    3      [ ]     STREAM   CONNECTED    30941  /run/dbus/system_bus_socket
unix    3      [ ]     STREAM   CONNECTED     22696
unix    3      [ ]     STREAM   CONNECTED     37476
unix    3      [ ]     STREAM   CONNECTED     34765  /run/user/1000/bus
unix    3      [ ]     STREAM   CONNECTED     35951
unix    3      [ ]     STREAM   CONNECTED     34227
unix    3      [ ]     STREAM   CONNECTED     34151
unix    3      [ ]     STREAM   CONNECTED     34086
unix    3      [ ]     STREAM   CONNECTED     21864

```



```

robin@robin-VirtualBox: $ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:domain          0.0.0.0:*
tcp      0      0 localhost:ipp             0.0.0.0:*
tcp6     0      0 ip6-localhost:ipp        [::]:*
robin@robin-VirtualBox: $ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp      0      0 0.0.0.0:49578            0.0.0.0:*
udp      0      0 0.0.0.0:631             0.0.0.0:*
udp      0      0 localhost:domain         0.0.0.0:*
udp      0      0 robin-VirtualBox:bootpc _gateway:bootps    ESTABLISHED
udp      0      0 0.0.0.0:mdns            0.0.0.0:*
udp6     0      0 [::]:59871              [::]:*
udp6     0      0 [::]:mdns              [::]:*
robin@robin-VirtualBox: $ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:domain          0.0.0.0:*
tcp      0      0 localhost:ipp             0.0.0.0:*
tcp6     0      0 ip6-localhost:ipp        [::]:*
udp      0      0 0.0.0.0:49578            0.0.0.0:*
udp      0      0 0.0.0.0:631             0.0.0.0:*
udp      0      0 localhost:domain         0.0.0.0:*
udp      0      0 0.0.0.0:mdns            0.0.0.0:*
robin@robin-VirtualBox: ~$ 

robin@robin-VirtualBox: ~$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:domain          0.0.0.0:*
tcp      0      0 localhost:ipp             0.0.0.0:*
tcp6     0      0 ip6-localhost:ipp        [::]:*
robin@robin-VirtualBox: ~$ 

```

If config

- The if config commands is used for displaying current network configuration information , setting up an ip address , netmask or broadcast address to an network interface , creating an alias for network interface , setting up hardware address and enable or disable network interface
- ifconfig –a :This option is used to display all the interfaces available, even if they are down.
- ifconfig -s : Display a short list, instead of details
- ifconfig interface up :This option is used to activate the driver for the given interface
- ifconfig interface down :This option is used to deactivate the driver for the given interface.

```

robin@robin-VirtualBox:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
                inet6 fe80::8176:5957:a92:9cf prefixlen 64 scopeid 0x20<link>
                  ether 08:00:27:92:e9:f9 txqueuelen 1000 (Ethernet)
                    RX packets 394 bytes 226912 (226.9 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 433 bytes 37212 (37.2 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                  loop txqueuelen 1000 (Local Loopback)
                    RX packets 317 bytes 26654 (26.6 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 317 bytes 26654 (26.6 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

robin@robin-VirtualBox:~$ ifconfig -s
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500      394      0      0 0       433      0      0      0 BMRU
lo        65536      317      0      0 0       317      0      0      0 LRU
robin@robin-VirtualBox:~$ 

```

Nslookup

Command	Used for
nslookup -type=any google.com	Lookup for any record
nslookup -type=soa redhat.com	Lookup for an soa record(start of authority)
nslookup -type=ns google.com	Lookup for an ns record(Name Server)
nslookup -type=mx google.com	Lookup for an mx record(Mail Exchange)
nslookup -type=txt google.com	Lookup for an txt record

```
robin@robin-VirtualBox:~$ nslookup google.com
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: google.com
Address: 142.250.182.78
Name: google.com
Address: 2404:6800:4007:81b::200e

robin@robin-VirtualBox:~$ nslookup 10.0.2.15
15.2.0.10.in-addr.arpa name = robin-VirtualBox.
15.2.0.10.in-addr.arpa name = robin-VirtualBox.local.

Authoritative answers can be found from:

robin@robin-VirtualBox:~$
```

```
robin@robin-VirtualBox:~$ nslookup -type=any google.com
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: google.com
Address: 142.250.182.78
Name: google.com
Address: 2404:6800:4009:830::200e
google.com mail exchanger = 40 alt3.aspmx.l.google.com.
google.com mail exchanger = 20 alt1.aspmx.l.google.com.
google.com mail exchanger = 10 aspmx.l.google.com.
google.com mail exchanger = 30 alt2.aspmx.l.google.com.
google.com mail exchanger = 50 alt4.aspmx.l.google.com.
google.com origin = ns1.google.com
mail addr = dns-admin.google.com
serial = 394418365
refresh = 900
retry = 900
expire = 1800
minimum = 60
google.com nameserver = ns3.google.com.
google.com nameserver = ns1.google.com.
google.com nameserver = ns4.google.com.
google.com nameserver = ns2.google.com.

Authoritative answers can be found from:

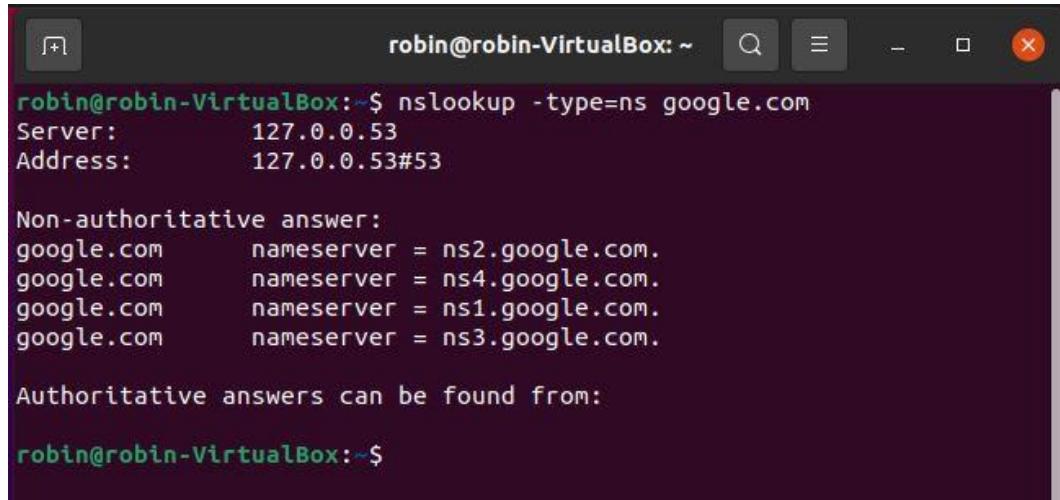
robin@robin-VirtualBox:~$
```

```
robin@robin-VirtualBox:~$ nslookup -type=soa redhat.com
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
redhat.com
origin = a1-68.akam.net
mail addr = noc.redhat.com
serial = 2021090302
refresh = 300
retry = 180
expire = 604800
minimum = 14400

Authoritative answers can be found from:

robin@robin-VirtualBox:~$
```



```
robin@robin-VirtualBox:~$ nslookup -type=ns google.com
Server: 127.0.0.53
Address: 127.0.0.53#53

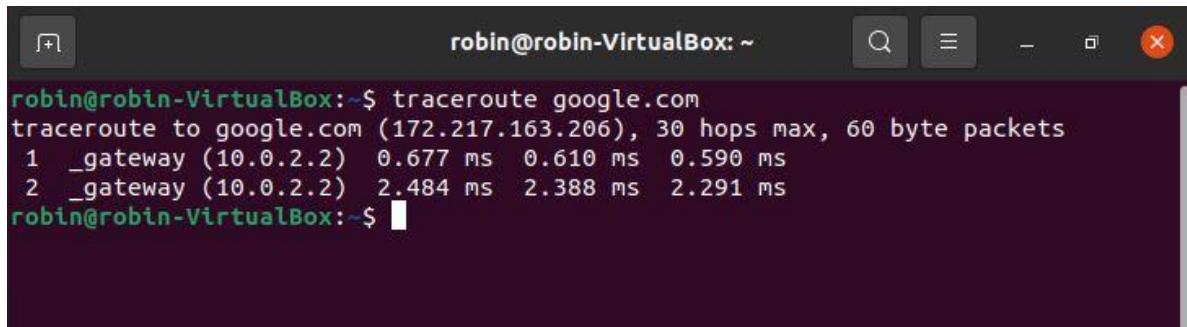
Non-authoritative answer:
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns3.google.com.

Authoritative answers can be found from:

robin@robin-VirtualBox:~$
```

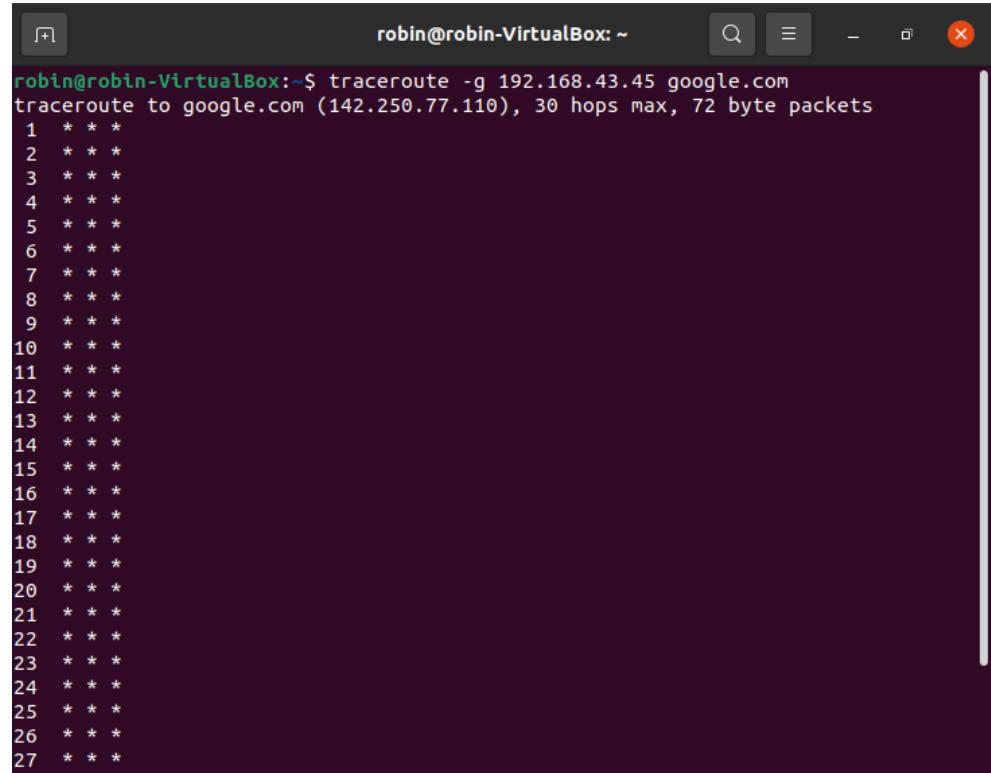
Traceroute

- **traceroute** command prints the route that a packet takes to reach the host. This command is useful when we want to know about the route and about all the hops that a packet takes

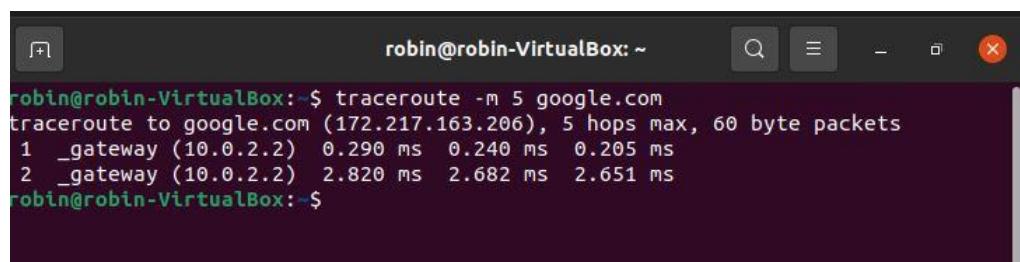


```
robin@robin-VirtualBox:~$ traceroute google.com
traceroute to google.com (172.217.163.206), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.677 ms  0.610 ms  0.590 ms
 2 _gateway (10.0.2.2)  2.484 ms  2.388 ms  2.291 ms
robin@robin-VirtualBox:~$
```

Command	Used for
\$ traceroute -g 192.168.43.45 google.com	Route the packet through gate
\$traceroute -m 5 google.com	Set the max number of hops for the packet to reach the destination. Default value is 30.
\$traceroute -n google.com	Do not resolve IP addresses to their domain names
\$traceroute -p 20292 google.com	Set the destination port to use. Default is 33434



```
robin@robin-VirtualBox:~$ traceroute -g 192.168.43.45 google.com
traceroute to google.com (142.250.77.110), 30 hops max, 72 byte packets
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 * * *
```



```
robin@robin-VirtualBox:~$ traceroute -m 5 google.com
traceroute to google.com (172.217.163.206), 5 hops max, 60 byte packets
1 _gateway (10.0.2.2) 0.290 ms 0.240 ms 0.205 ms
2 _gateway (10.0.2.2) 2.820 ms 2.682 ms 2.651 ms
robin@robin-VirtualBox:~$
```

\$route	To display the IP/kernel routing table
\$route -n	To display routing table in full numeric form
\$sudo route add default gw 169.254.0.0	To add a default gateway.
\$route -Cn	To list kernel's routing cache information.
\$sudo route add -host 192.168.1.51 reject	. To reject routing to a particular host or network
\$ip route	To get details of the kernel/IP routing table using ip command
\$route del default	To delete the default gateway.
\$ip route show table local	To get the details of the local table with destination addresses assigned to the localhost.

EXPERIMENT NO: 18

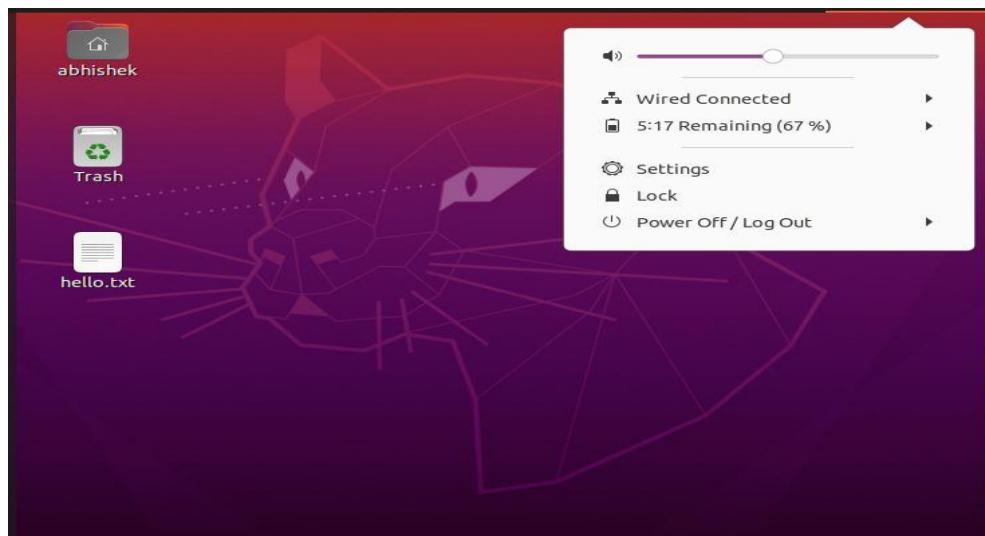
FAMILIARISATION OF STATIC AND DYNAMIC IP

STATIC AND DYNAMIC IP

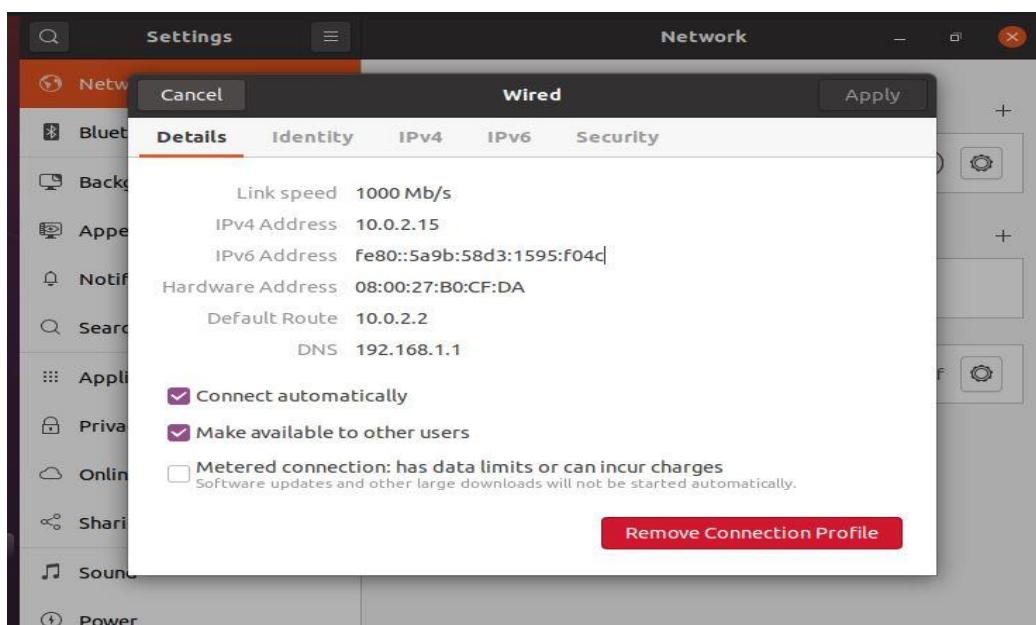
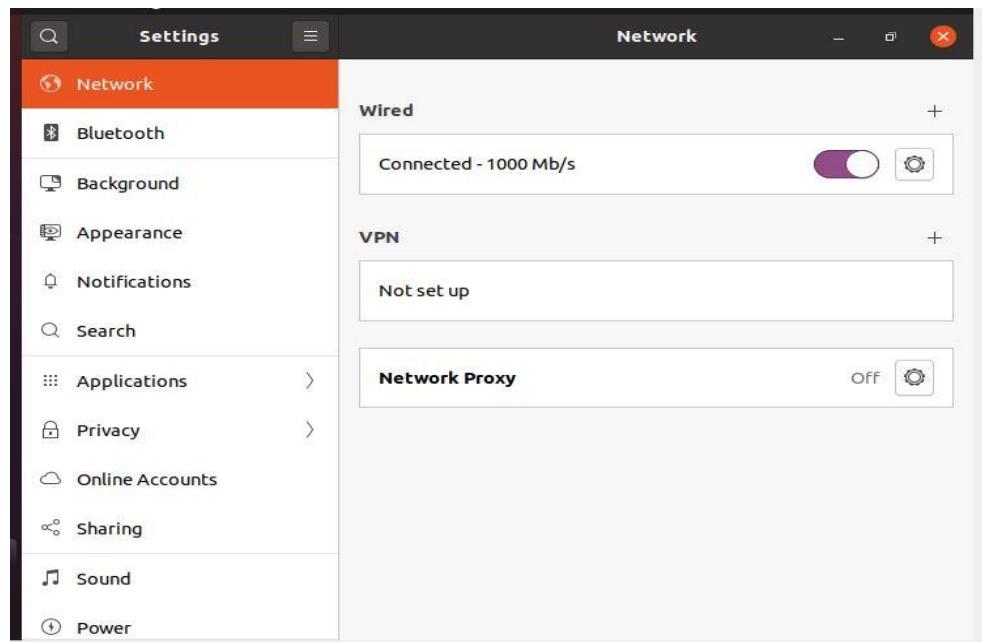
- A fixed IP address is called static IP address , i.e. it never changes.
- It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.
- It is used to connect to an IP camera, home file server, game server, and many other devices.
- A static IP address is necessary only for servers and not for personal PCs

STEPS FOR SETTING STATIC IP

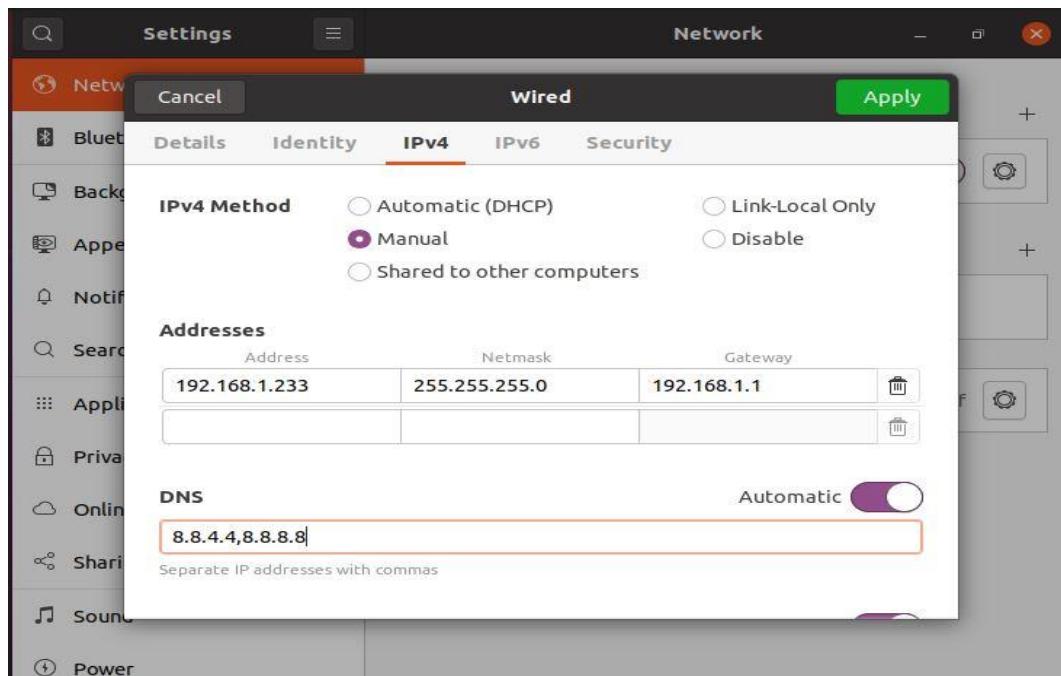
- *Click on the top right network icon and select settings of the network interface you wish to configure to use a static IP address on Ubuntu*



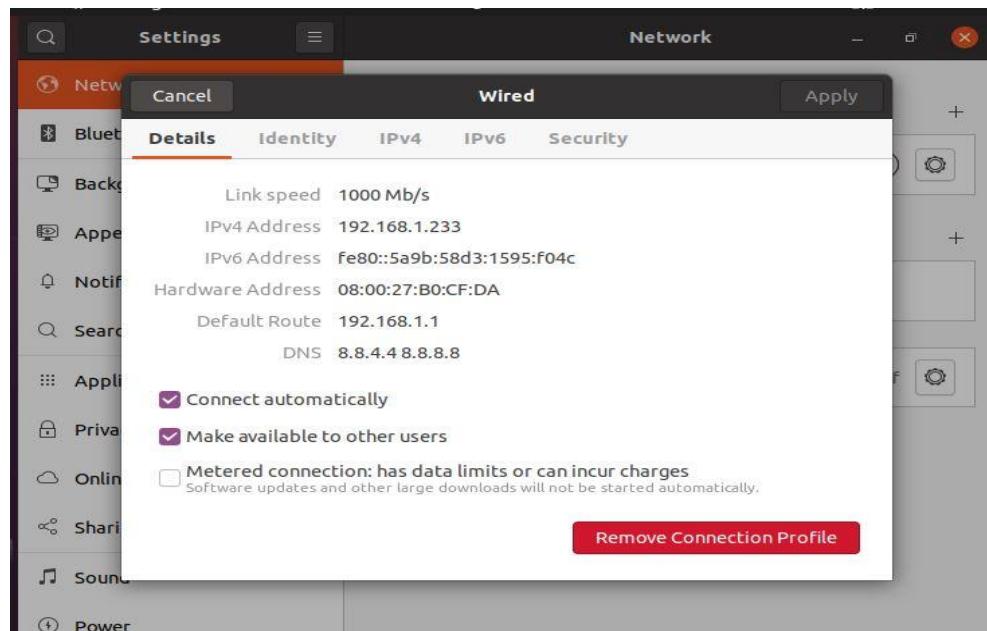
Click on the settings icon to start IP address configuration



- ❖ Select **IPv4** tab
- ❖ Select manual and enter your desired IP address, netmask, gateway and DNS settings.
Once ready click **Apply** button



- ❖ Turn OFF and ON switch to apply your new network static IP configuration settings.
- ❖ Click on the network settings icon once again to confirm your new static IP address settings.



DYNAMIC IP

- A dynamic IP address as its name suggests is a temporary IP address assigned by a DHCP server for every new network.
- A dynamic IP address is used due to the shortage of IP addresses on IPV4.
- A single dynamic IP address can be used between many devices

Configuring a dynamic ip address

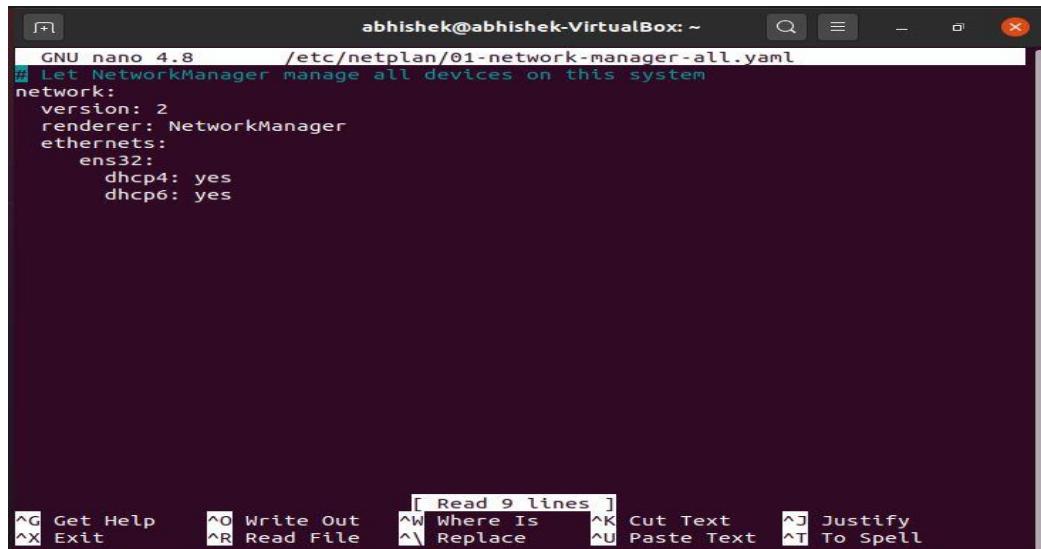
Step1: type the command in the terminal

sudo nano /etc/netplan/01-network-manager-all.yaml

```
robin@robin-VirtualBox:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
[sudo] password for robin:
```

step2: Now find the name of the network interface you want to configure and insert the following lines:

- **dhcp4: yes**
- **dhcp6: yes**



```
GNU nano 4.8      /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens32:
      dhcp4: yes
      dhcp6: yes
```

The terminal shows the nano editor with the following command-line options at the bottom:

- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste Text
- ^T To Spell

Step3: Apply the changes with **sudo netplan apply** command

```
robin@robin-VirtualBox:~$ sudo netplan apply
robin@robin-VirtualBox:~$
```

EXPERIMENT NO-19**CONCEPT OF SUBNETS AND CIDR ADDRESS SCHEME****SUBNETS:**

The process of dividing a network into smaller network sections is called subnetting. This can be useful for many different purposes and helps isolate groups of hosts from each other to deal with them more easily. By default, each network has only one subnet, which contains all of the host addresses defined within. A netmask is basically a specification of the amount of address bits that are used for the network portion. A subnet mask is another netmask within used to further divide the network.

Each bit of the address that is considered significant for describing the network should be represented as a “1” in the netmask. For instance, the address we discussed above, 192.168.0.15 can be expressed like this, in binary:

1100 0000 - 1010 1000 - 0000 0000 - 0000 1111

As we described above, the network portion for class C addresses is the first 3 octets, or the first 24 bits. Since these are the significant bits that we want to preserve, the netmask would be:

1111 1111 - 1111 1111 - 1111 1111 - 0000 0000

This can be written in the normal IPv4 format as 255.255.255.0. Any bit that is a “0” in the binary representation of the netmask is considered part of the host portion of the address and can be variable. The bits that are “1” are static, however, for the network or subnetwork that is being discussed. We determine the network portion of the address by applying a bitwise AND operation to between the address and the netmask. A bitwise AND operation will save the networking portion of the address and discard the host portion. The result of this on our above example that represents our network is:

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

This can be expressed as 192.168.0.0. The host specification is then the difference between these original values and the host portion. In our case, the host is 0000 1111 or 15. The idea of subnetting is to take a portion of the host space of an address, and use it as an additional networking specification

to divide the address space again. For instance, a netmask of 255.255.255.0 as we saw above leaves us with 254 hosts in the network (you cannot end in 0 or 255 because these are reserved).

So, continuing with our example, the networking portion is: 1100 0000 - 1010 1000 - 0000 0000

The host portion is:

0000 1111

We can use the first bit of our host to designate a subnetwork. We can do this by adjusting the subnetmask from this:

1111 1111 - 1111 1111 - 1111 1111 - 0000 0000

To this:

1111 1111 - 1111 1111 - 1111 1111 - 1000 0000

In traditional IPv4 notation, this would be expressed as 192.168.0.128. What we have done here is to designate the first bit of the last octet as significant in addressing the network. This effectively produces two subnetworks. The first subnetwork is from 192.168.0.1 to 192.168.0.127. The second subnetwork contains the hosts 192.168.0.129 to 192.168.0.255.

CIDR NOTATION:

A system called Classless Inter-Domain Routing, or CIDR, was developed as an alternative to traditional subnetting. For example, we could express the idea that the IP address 192.168.0.15 is associated with the netmask 255.255.255.0 by using the CIDR notation of 192.168.0.15/24. This means that the first 24 bits of the IP address given are considered significant for the network routing.

This allows us some interesting possibilities. We can use these to reference “supernets”. In this case, we mean a more inclusive address range that is not possible with a traditional subnet mask. For instance, in a class C network, like above, we could not combine the addresses from the networks 192.168.0.0 and 192.168.1.0 because the netmask for class C addresses is 255.255.255.0. However, using CIDR notation, we can combine these blocks by referencing this chunk as 192.168.0.0/23. This specifies that there are 23 bits used for the network portion that we are referring to. So the first network (192.168.0.0) could be represented like this in binary:

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

While the second network (192.168.1.0) would be like this:

1100 0000 - 1010 1000 - 0000 0001 - 0000 0000

The CIDR address we specified indicates that the first 23 bits are used for the network block we are referencing. This is equivalent to a netmask of 255.255.254.0, or:

1111 1111 - 1111 1111 - 1111 1110 - 0000 0000

As you can see, with this block the 24th bit can be either 0 or 1 and it will still match, because the network block only cares about the first 23 digits. CIDR allows us more control over addressing continuous blocks of IP addresses. This is much more useful than the subnetting we talked about originally.

EXPERIMENT NO- 20**CONCEPT OF SUBNET MASK**

The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network. In TCP/IP, the parts of the IP address that are used as the network and host addresses aren't fixed. Unless you have more information, the network and host addresses above can't be determined. This information is supplied in another 32-bit number called a subnet mask. The subnet mask is 255.255.255.0 in this example. It isn't obvious what this number means unless you know 255 in binary notation equals 11111111. So, the subnet mask is 11111111.11111111.11111111.00000000. Lining up the IP address and the subnet mask together, the network, and host portions of the address can be separated:

11000000.10101000.0111011.10000100 - IP address (192.168.123.132)

11111111.11111111.11111111.00000000 - Subnet mask (255.255.255.0)

The first 24 bits (the number of ones in the subnet mask) are identified as the network address. The last 8 bits (the number of remaining zeros in the subnet mask) are identified as the host address. It gives you the following addresses:

11000000.10101000.0111011.00000000 - Network address (192.168.123.0)

00000000.00000000.00000000.10000100 - Host address (0.0.0.132)

So now you know, for this example using a 255.255.255.0 subnet mask, that the network ID is 192.168.123.0, and the host address is 0.0.0.132. When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it. Almost all decimal subnet masks convert to binary numbers that are all ones on the left and all zeros on the right.

Some other common subnet masks are:

Decimal Binary 255.255.255.192 1111111.11111111.11111111.11000000 255.255.255.224

11111111.11111111.11111111.11100000

Internet RFC 1878 (available from InterNIC-Public Information Regarding Internet Domain Name Registration Services) describes the valid subnets and subnet masks that can be used on TCP/IP networks

EXPERIMENT NO: 21**SETTING UP A FIREWALL FOR LAN****ufw-uncomplicated firewall**

- ❑ The default firewall configuration tool for Ubuntu is ufw.
- ❑ Developed to ease iptables firewall configuration.
- ❑ ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.
- ❑ ufw by default is initially disabled.
- ❑ ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules.
- ❑ It is currently mainly used for host-based firewalls.”

To ENABLE ufw

- ❖ Step1:check current firewall status

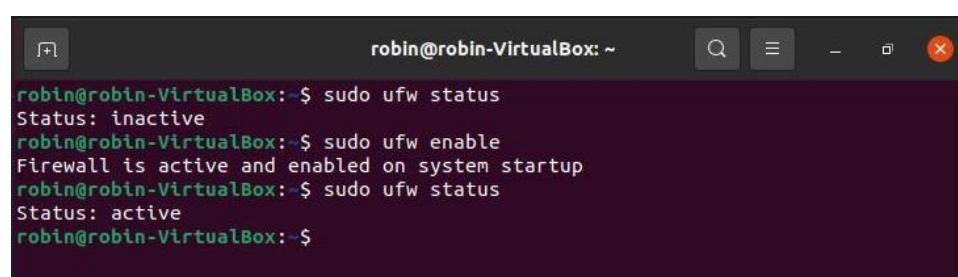
sudo ufw status

- ❖ Step2:to enable firewall

sudo ufw enable

- ❖ Step3:to check status

sudo ufw status



```
robin@robin-VirtualBox:~$ sudo ufw status
Status: inactive
robin@robin-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
robin@robin-VirtualBox:~$ sudo ufw status
Status: active
robin@robin-VirtualBox:~$
```

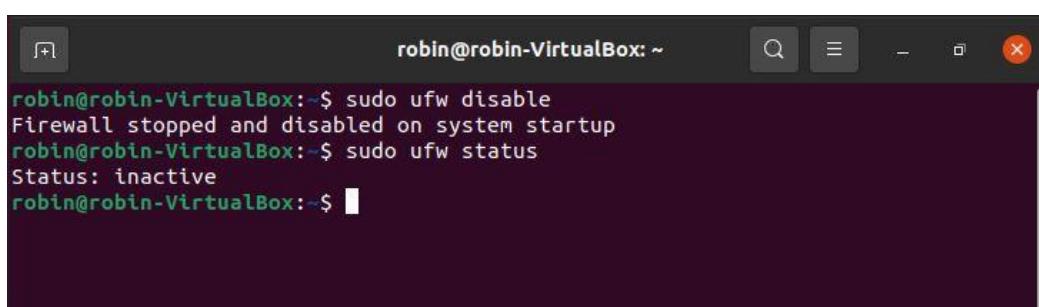
To Disable ufw

- ❖ Step1:To disable firewall

sudo ufw disable

- ❖ Step2:to check status

sudo ufw status



```
robin@robin-VirtualBox:~$ sudo ufw disable
Firewall stopped and disabled on system startup
robin@robin-VirtualBox:~$ sudo ufw status
Status: inactive
robin@robin-VirtualBox:~$
```

EXPERIMENT NO: 22

WIRESHARK AND TCPDUMP

WHAT IS WIRESHARK?

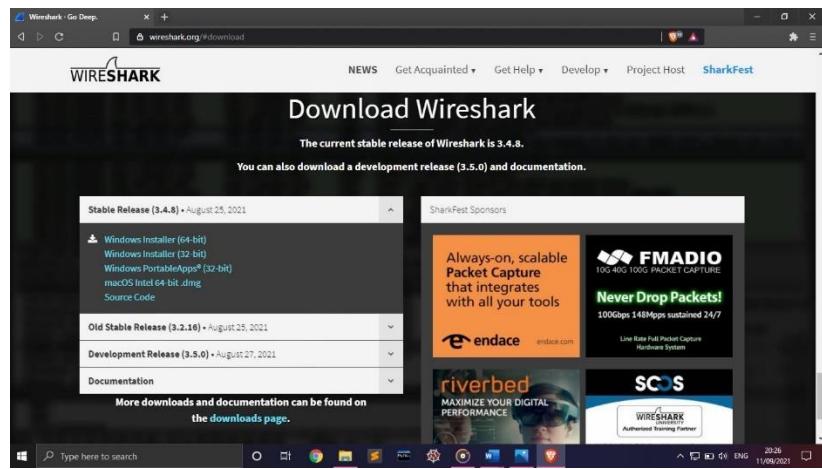
- Network packet protocol analyzer
- A network packet analyzer will try to capture network packets and try to display that packet data as detailed as possible.
- One of the best open source packet analyzers available today for UNIX and Windows

WHERE IT USE ?

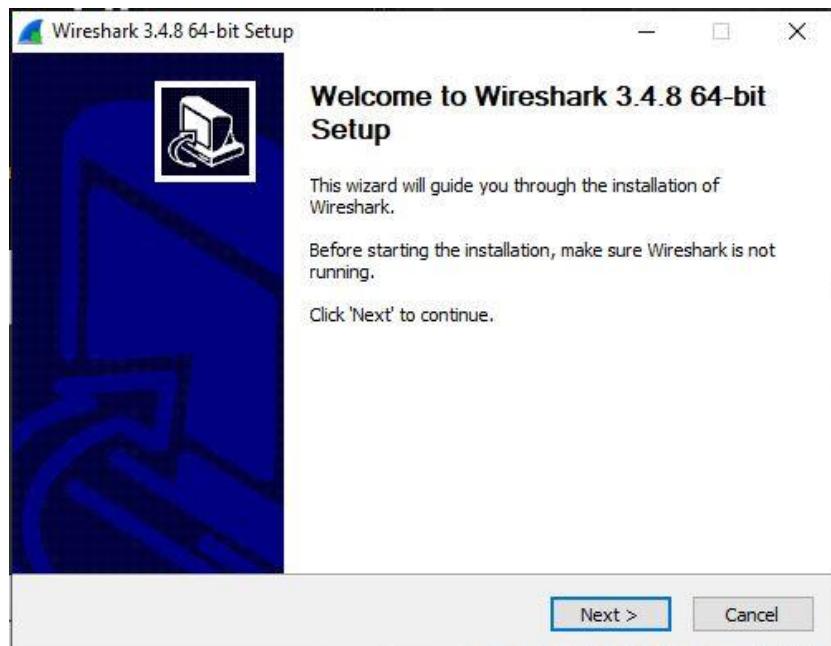
- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- Testers use it to detect defects
- People use it to learn network protocol internals

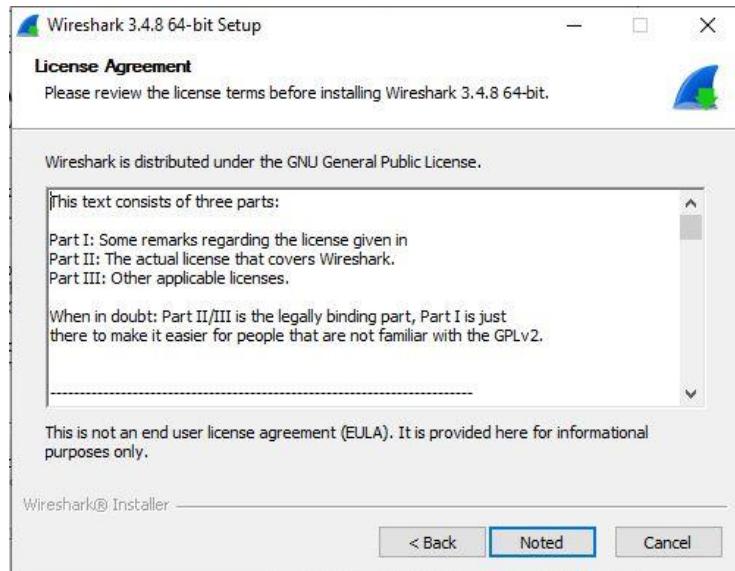
STEPS TO INSTALL WIRESHARK

Step1: Go to www.wireshark.org ->download 64-bit package

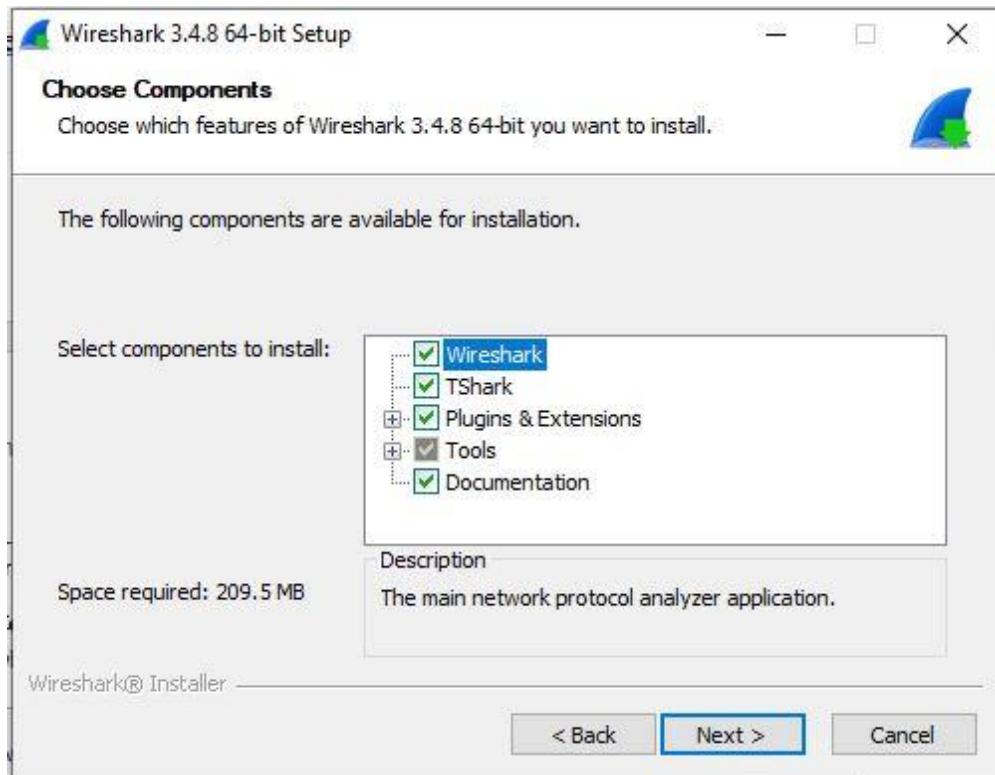


Step 2 : run application and click on noted

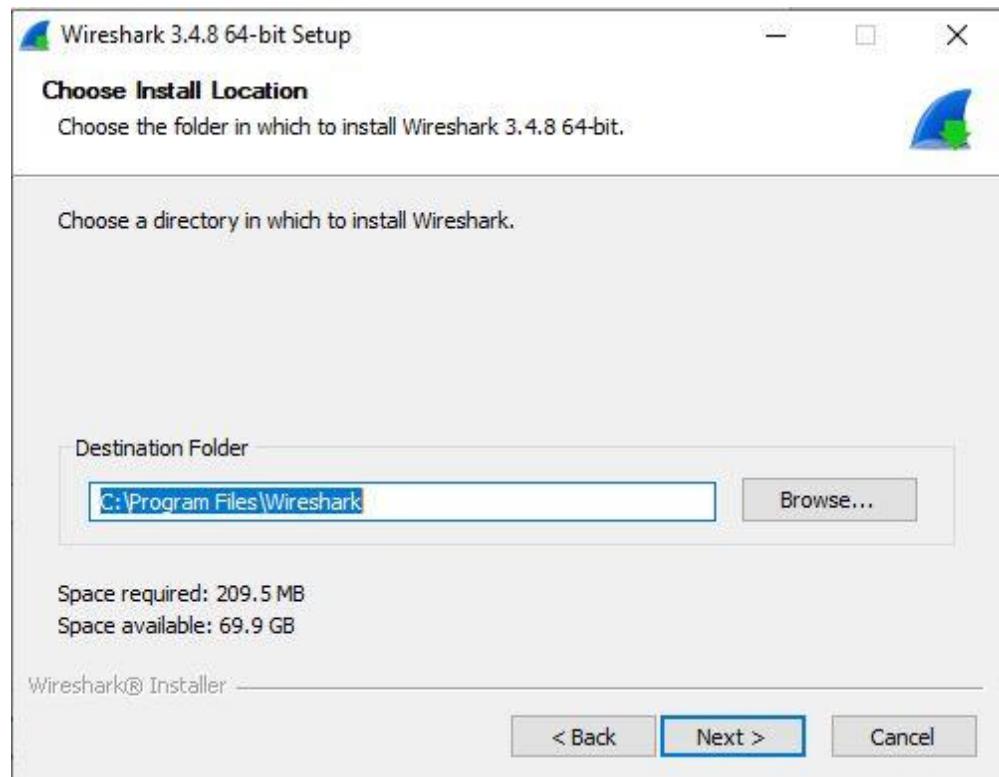




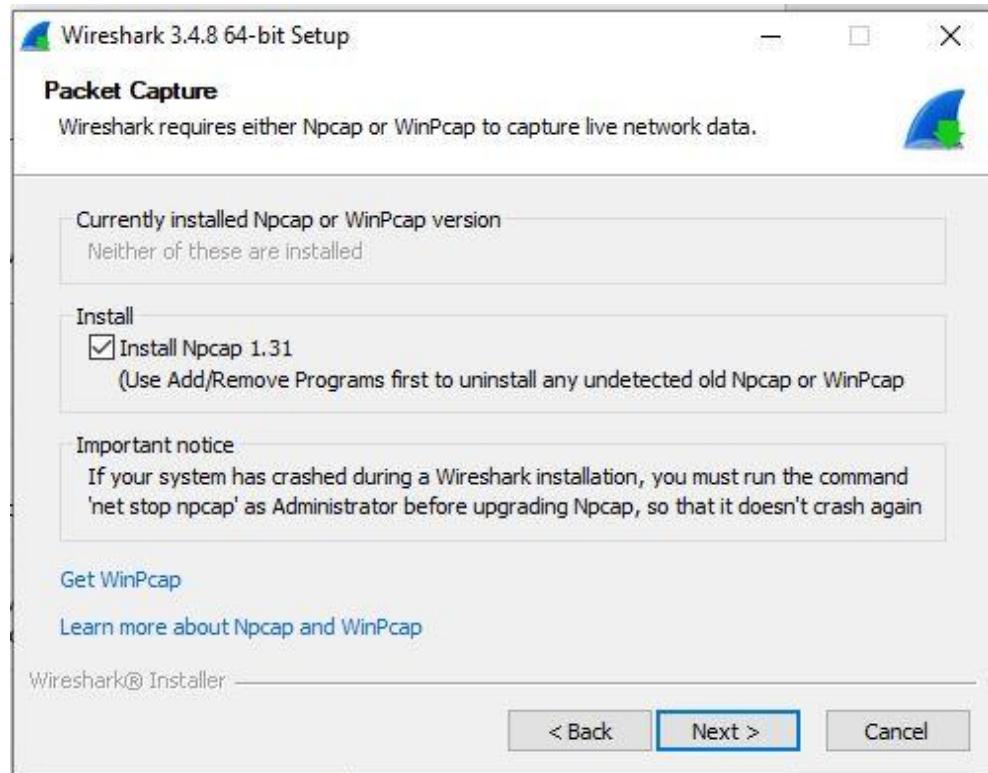
Step 3 :select components and click next

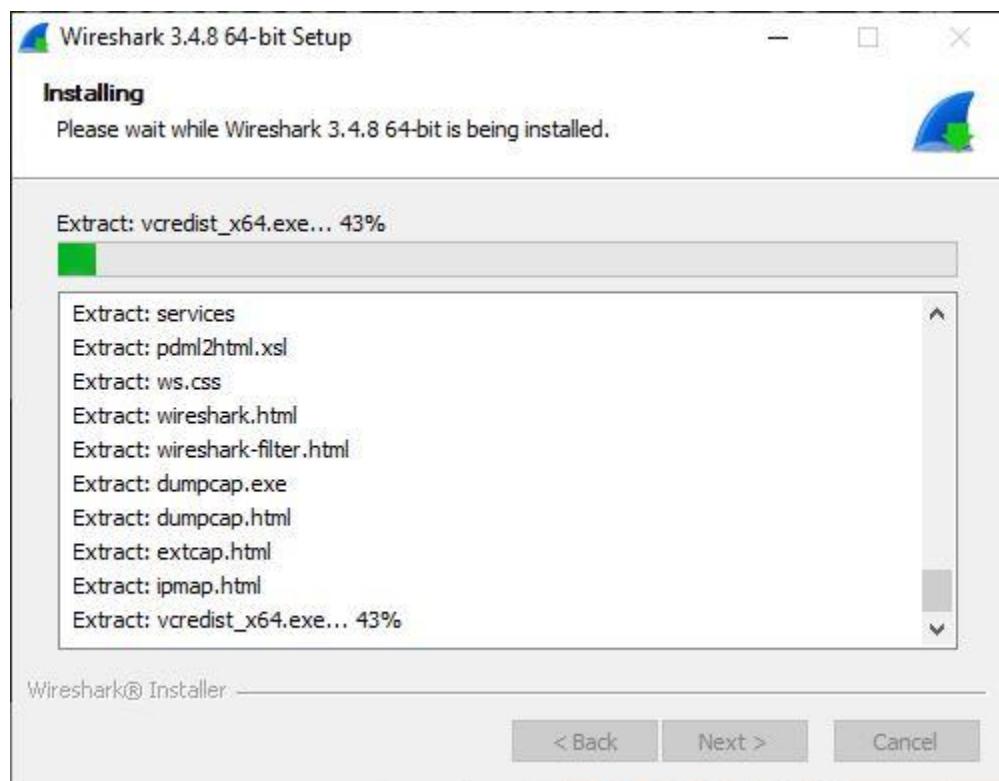
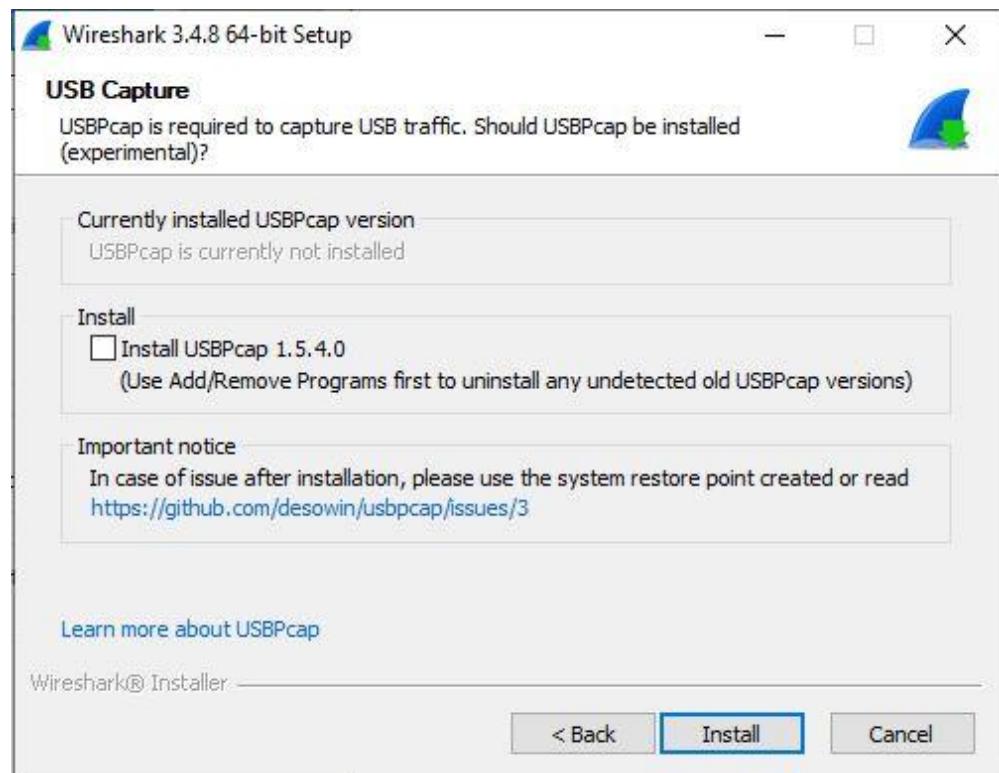


Step 4: choose default destination location

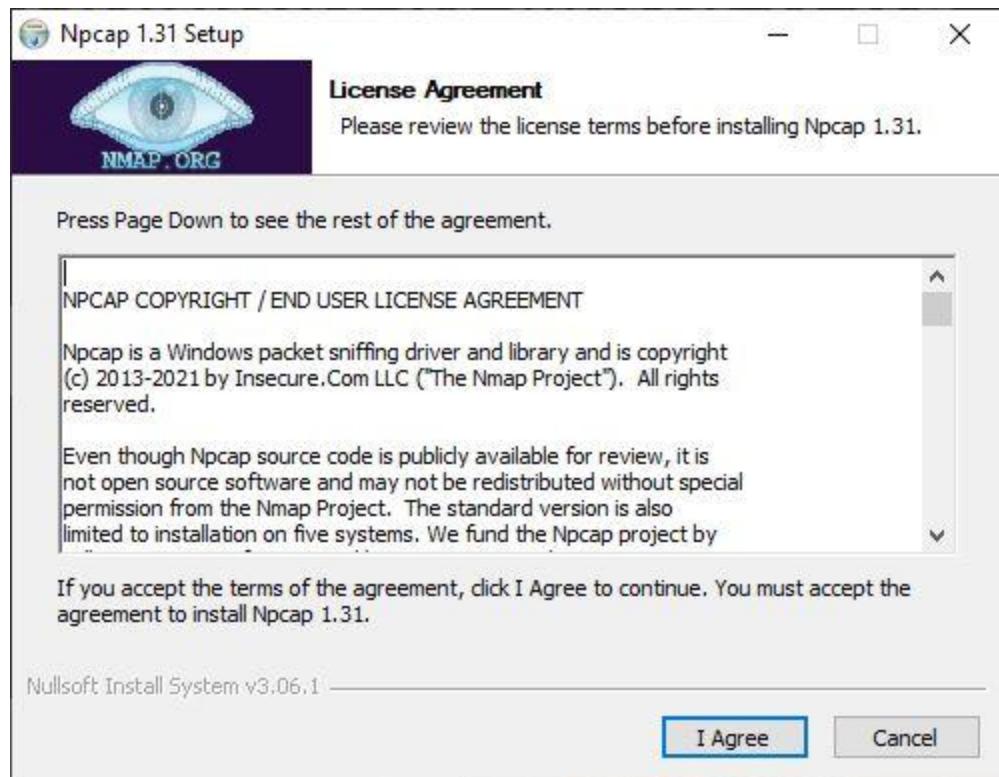


Step 5: installation start running

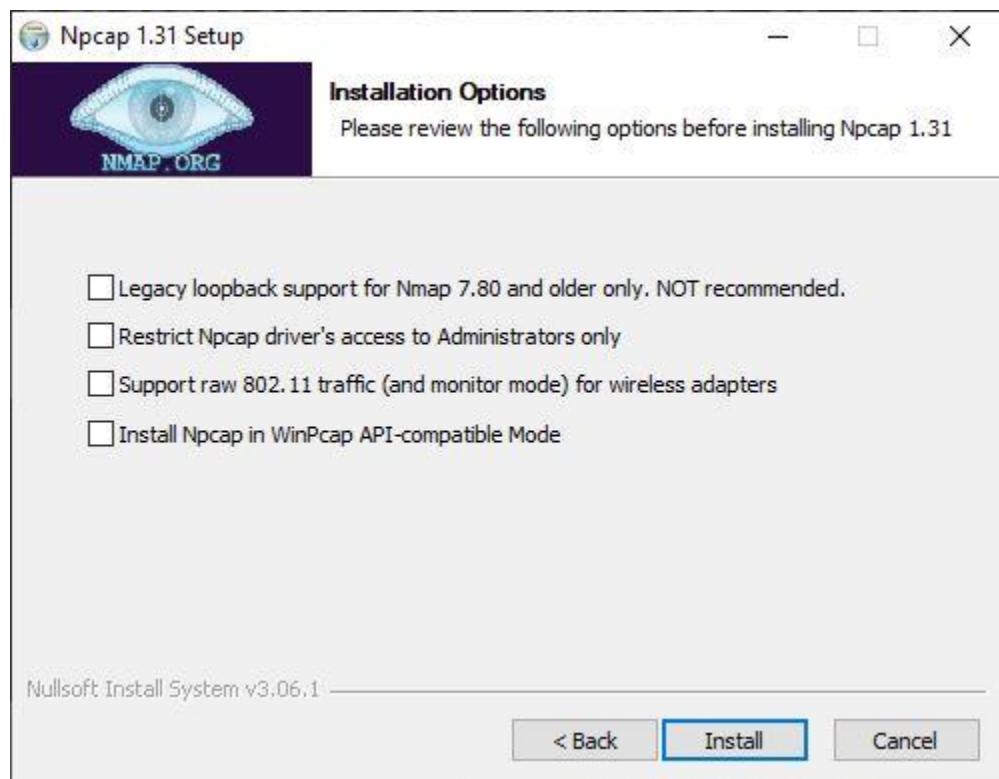


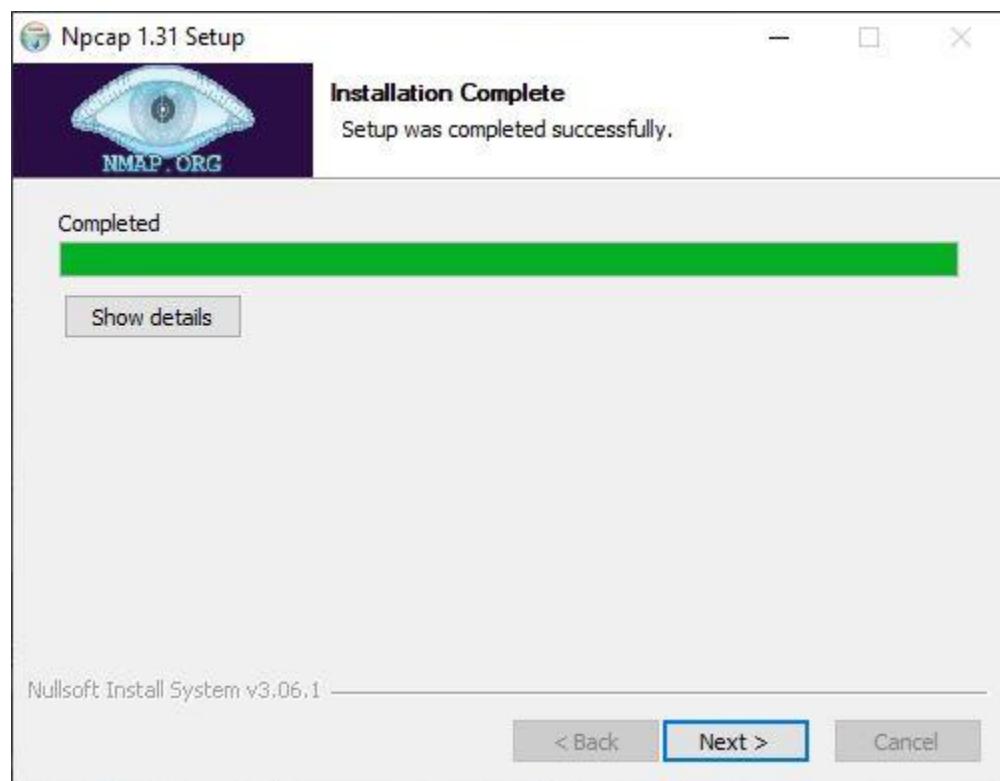
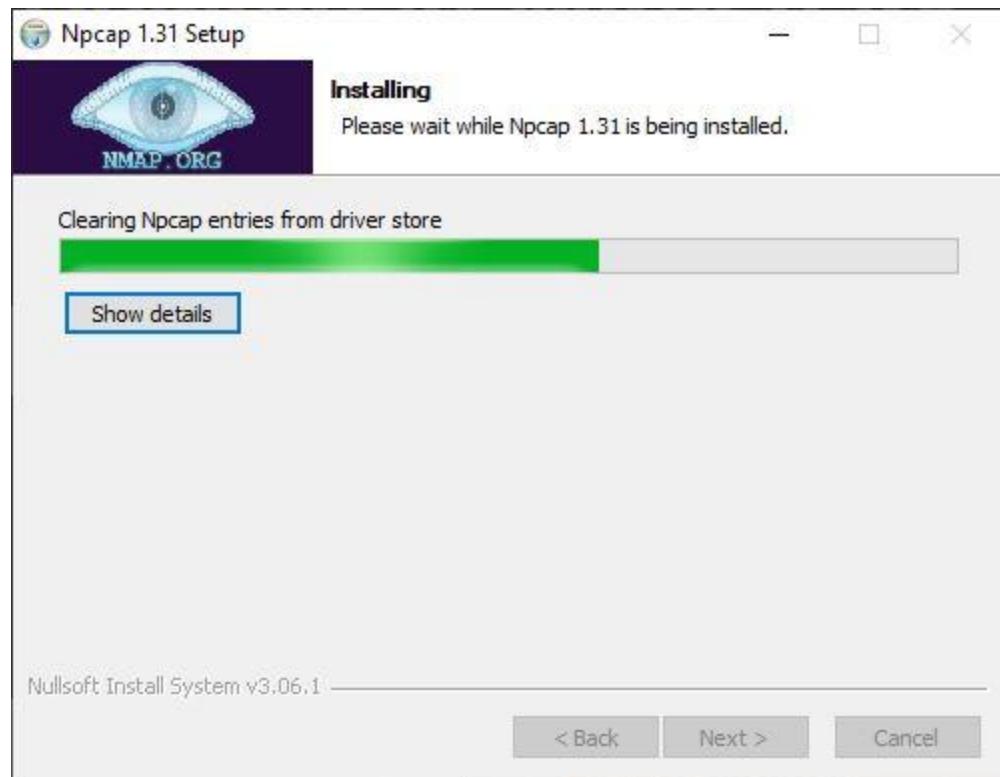


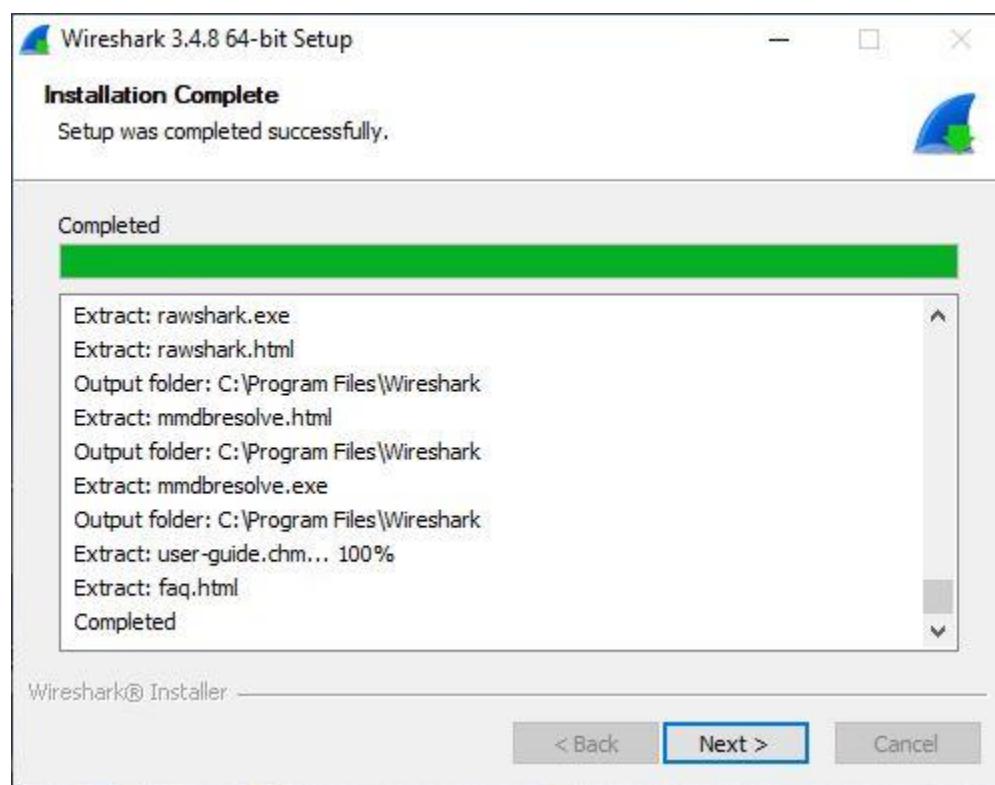
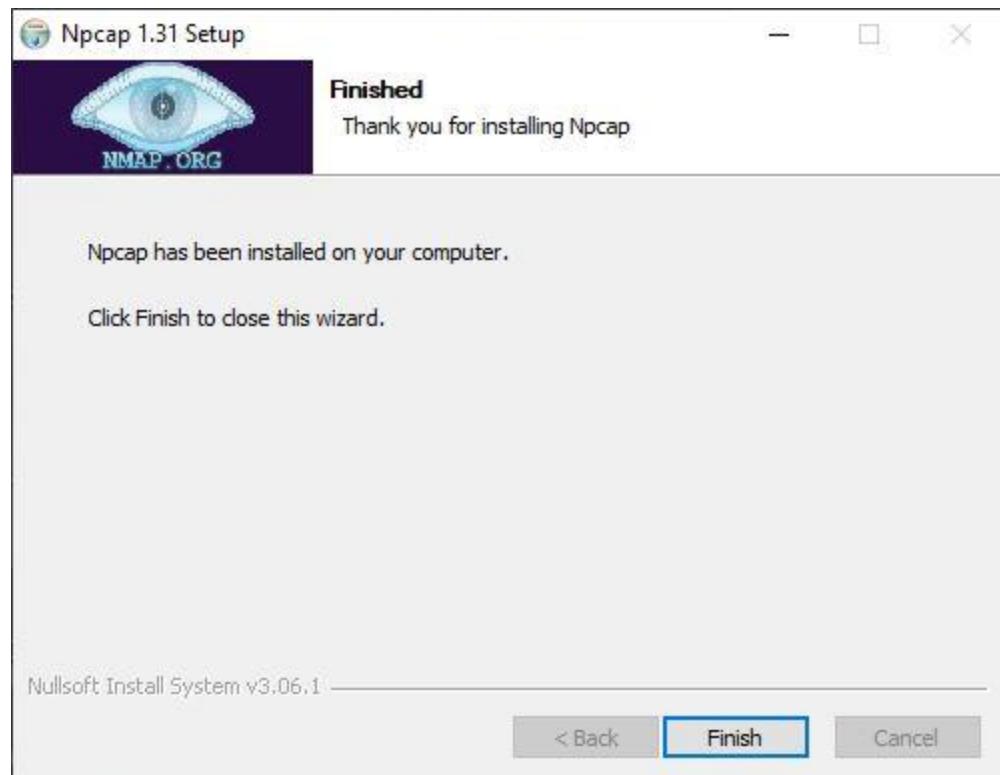
step 6 : click on I agree



Step 7 : complete installation part and click finish

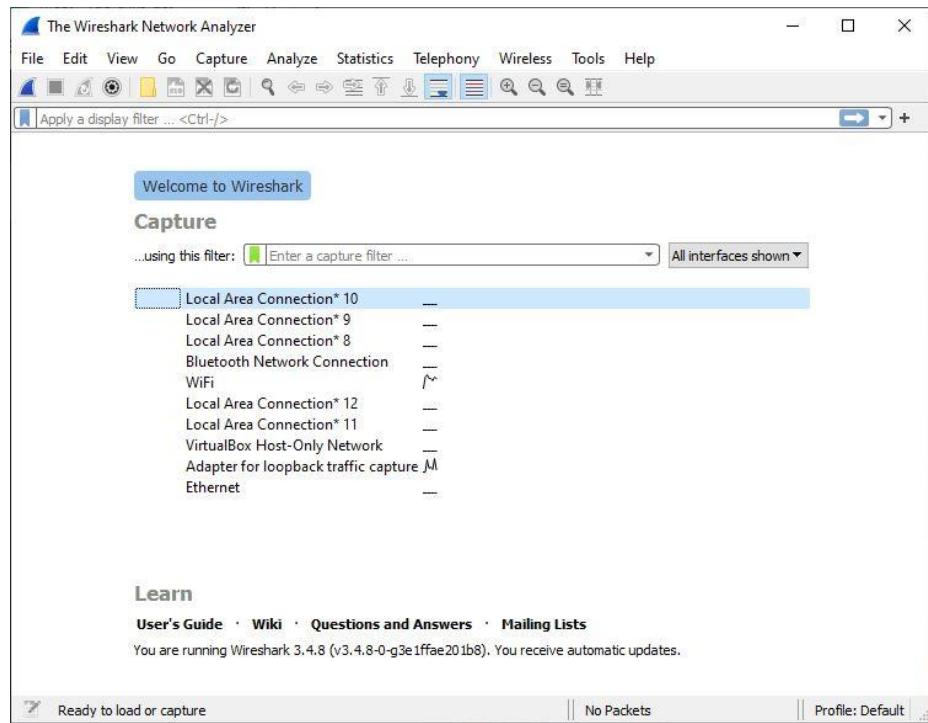






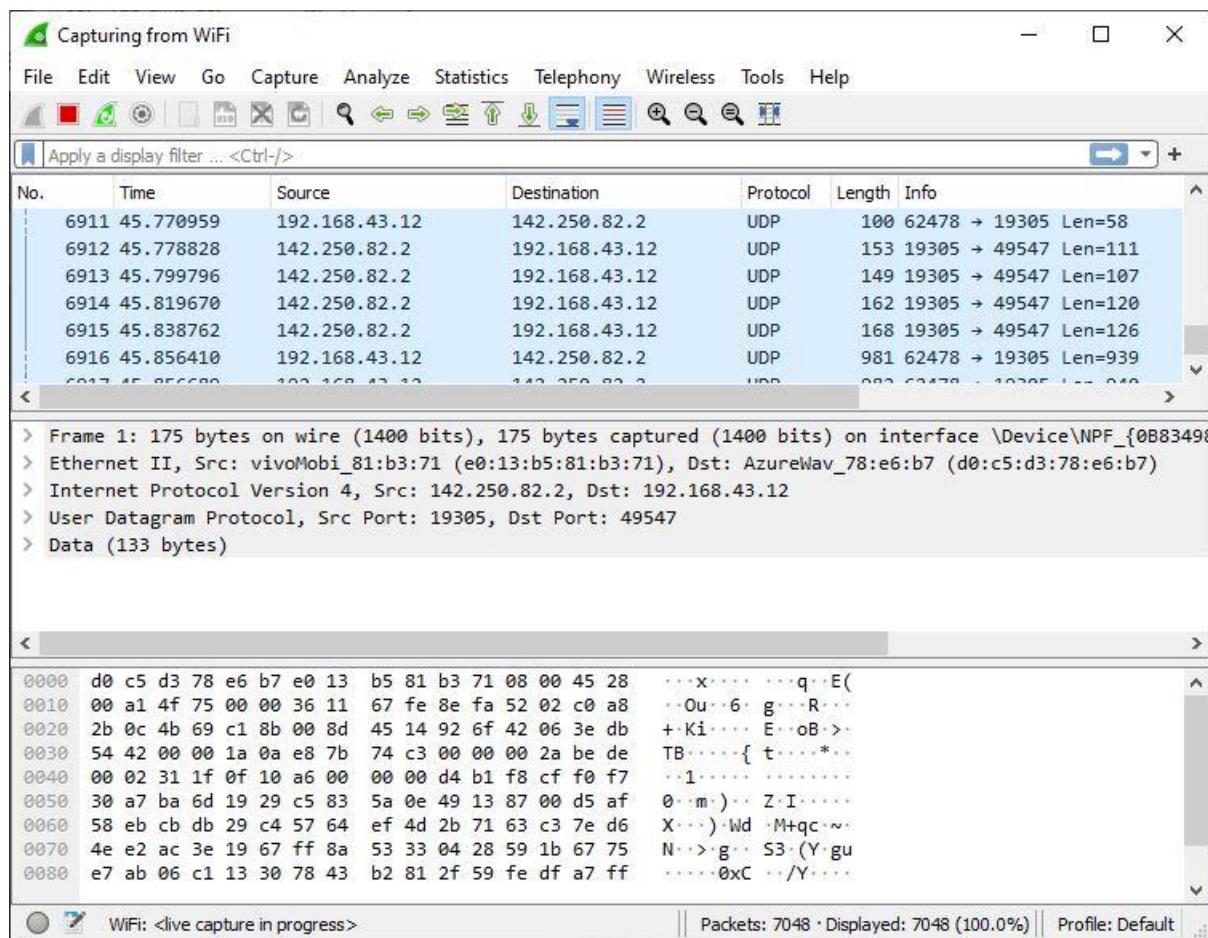


HOMESCREEN



HOW IT CAPTURE PACKETS?

- Wireshark captures packets and lets you examine their contents
- Select any interface to capture its packet



- No. shows the number of captured packet or index number.
- Time shows the time of capture
- Source shows the source ip of the packet or the packet is originally generated from which source ip.
- Destination shows the destination ip where the packet is going.
- Protocol shows which kind of protocol communication is held between the source and destination.
- Info shows the data payload in the packet

FEATURES OF WIRESHARK

- Available for UNIX and Windows.

- Capture live packet data from a network interface.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

TCPDUMP

- It is an ip utility tool used for real-time packet sniffing(Network).
- Command line program comes in built in a Unix based system.
- Programs like ethereal(Wireshark) provide an alternative to Tcpdump in GUI environment

STEPS TO INSTALL TCPDUMP

- Install tcpdump by entering the following commands in the terminal:

sudo apt update

```
robin@robin-VirtualBox:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu groovy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu groovy-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu groovy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu groovy-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
320 packages can be upgraded. Run 'apt list --upgradable' to see them.
robin@robin-VirtualBox:~$ █
```

sudo apt install tcpdump

```
robin@robin-VirtualBox:~$ sudo apt install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-6).
tcpdump set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 320 not upgraded.
robin@robin-VirtualBox:~$
```

TCPDUMP COMMAND EXAMPLES

1. Display Available Interfaces

```
# tcpdump -D
```

```
robin@robin-VirtualBox:~$ tcpdump -D
1.enp0s3 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7 dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
robin@robin-VirtualBox:~$
```

2. Capture Packets from Specific Interface

```
# tcpdump -i any
```

- The command screen will scroll up until you interrupt and when we execute the tcpdump command it will captures from all the interfaces, however with -i switch only capture from the desired interface

3. Print Captured Packets in ASCII

- The below tcpdump command with the option -A displays the package in ASCII format. It is a character-encoding scheme format.

```
# tcpdump -A -i any
```

4. Capture Only N Number of Packets

- When you run the tcpdump command it will capture all the packets for the specified interface, until you hit the cancel button. But using -c option, you can capture a specified number of packets.

```
# tcpdump -c 5 -i any
```

5.Display Captured Packets in HEX and ASCII

- The following command with option -XX capture the data of each packet, including its link level header in HEX and ASCII format.

```
# tcpdump -XX -i any
```

6.Capture and Save Packets in a File

- As we said, that tcpdump has a feature to capture and save the file in a .Pcap format, to do this just execute the command with -w option.

```
# tcpdump -w 0001.Pcap -i any
```

Testing network services with Netcat [nc]

- Use the netcat command, nc, to access the service. If you don't have nc installed, type the following command on the command line

Step 1: \$ sudo apt-get install netcat

```
robin@robin-VirtualBox:~$ sudo apt-get install netcat
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  netcat
0 upgraded, 1 newly installed, 0 to remove and 320 not upgraded.
Need to get 2,184 B of archives.
After this operation, 15.4 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu groovy/universe amd64 netcat all 1.21
7-2ubuntu1 [2,184 B]
Fetched 2,184 B in 1s (4,307 B/s)
Selecting previously unselected package netcat.
(Reading database ... 191149 files and directories currently installed.)
Preparing to unpack .../netcat_1.217-2ubuntu1_all.deb ...
Unpacking netcat (1.217-2ubuntu1) ...
Setting up netcat (1.217-2ubuntu1) ...
robin@robin-VirtualBox:~$
```

Step 2: After the installation is done type 'nc -h'

```
robin@robin-VirtualBox:~$ nc -h
OpenBSD netcat (Debian patchlevel 1.217-2ubuntu1)
usage: nc [-46CddFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
  Command Summary:
    -4           Use IPv4
    -6           Use IPv6
    -b           Allow broadcast
    -C           Send CRLF as line-ending
    -D           Enable the debug socket option
    -d           Detach from stdin
    -F           Pass socket fd
    -h           This help text
    -I length   TCP receive buffer length
    -i interval  Delay interval for lines sent, ports scanned
    -k           Keep inbound sockets open for multiple connects
    -l           Listen mode, for inbound connects
    -M ttl      Outgoing TTL / Hop Limit
    -m minttl   Minimum incoming TTL / Hop Limit
    -N           Shutdown the network socket after EOF on stdin
    -n           Suppress name/port resolutions
    -O length   TCP send buffer length
    -P proxyuser Username for proxy authentication
    -p port     Specify local port for remote connects
    -q secs     quit after EOF on stdin and delay of secs
    -r           Randomize remote ports
```

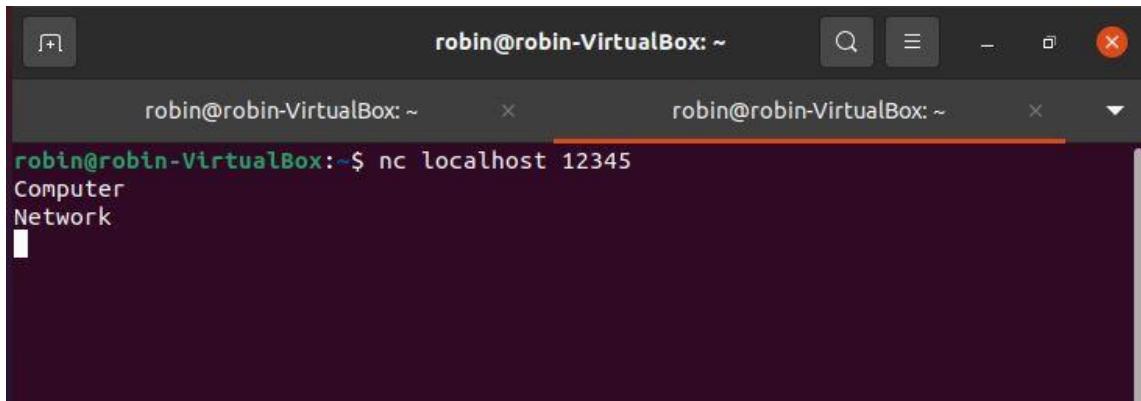
Step3: Set up the server using netcat in listening mode.

We will use port 12345 and will specify the port number with -p option.

```
robin@robin-VirtualBox:~$ nc -l -p 12345
Computer
Network
```

Step 4: Creating the server with netcat

- The command ‘nc hostname port’ puts netcat in client mode and connects to the specified hostname on the specified port. Open a new terminal window and type ‘nc localhost 12345’



```
robin@robin-VirtualBox: ~
robin@robin-VirtualBox: ~
robin@robin-VirtualBox:~$ nc localhost 12345
Computer
Network
[redacted]
```

Step 5 : Now that we are connected to the server we can start chatting

EXPERIMENT NO 23

Analyse Packets using wireshark

1. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window. Support your answer with an appropriate screenshot from your computer.

Ans: TCP, UDP, DNS, TLSV1.2, etc..

No.	Time	Source	Destination	Protocol	Length	Info
316	19:43:31.802690	192.168.223.128	52.139.153.205	TLSv1.2	270	Client Hello
317	19:43:31.803328	52.139.153.205	192.168.223.128	TCP	60	443 → 49742 [ACK] Seq=1 Ack=217 Win=64240 Len=0
318	19:43:31.881394	13.91.16.71	192.168.223.128	TLSv1.2	174	Change Cipher Spec, Encrypted Handshake Message, Application...
319	19:43:31.883274	192.168.223.128	13.91.16.71	TLSv1.2	141	Application Data
320	19:43:31.883446	192.168.223.128	13.91.16.71	TLSv1.2	335	Application Data
321	19:43:31.883532	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=457 Win=64240 Len=0
322	19:43:31.883612	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=738 Win=64240 Len=0
323	19:43:31.883748	192.168.223.128	13.91.16.71	TLSv1.2	92	Application Data
324	19:43:31.883953	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=776 Win=64240 Len=0
325	19:43:31.884052	192.168.223.128	13.91.16.71	TCP	14654	49741 → 443 [ACK] Seq=776 Ack=2484 Win=64120 Len=14600 [TCP ...
326	19:43:31.884470	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=2236 Win=64240 Len=0
327	19:43:31.884500	192.168.223.128	13.91.16.71	TLSv1.2	2974	Application Data [TCP segment of a reassembled PDU]
328	19:43:31.884626	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=3694 Win=64240 Len=0
329	19:43:31.884626	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=5156 Win=64240 Len=0
330	19:43:31.884648	192.168.223.128	13.91.16.71	TCP	5894	49741 → 443 [ACK] Seq=18296 Ack=2484 Win=64120 Len=5840 [TCP ...
331	19:43:31.884731	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=6616 Win=64240 Len=0
332	19:43:31.884749	192.168.223.128	13.91.16.71	TCP	2974	49741 → 443 [ACK] Seq=24136 Ack=2484 Win=64120 Len=2920 [TCP ...
333	19:43:31.884848	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=8076 Win=64240 Len=0
334	19:43:31.884848	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=9536 Win=64240 Len=0
335	19:43:31.884873	192.168.223.128	13.91.16.71	TLSv1.2	855	Application Data
336	19:43:31.884983	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=10996 Win=64240 Len=0
337	19:43:31.885176	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=12456 Win=64240 Len=0
338	19:43:31.885225	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=13916 Win=64240 Len=0
339	19:43:31.885225	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=15376 Win=64240 Len=0
340	19:43:31.885319	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=16836 Win=64240 Len=0
341	19:43:31.885319	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=18296 Win=64240 Len=0
342	19:43:31.885384	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=19756 Win=64240 Len=0
343	19:43:31.885384	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=21216 Win=64240 Len=0
344	19:43:31.885452	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=22676 Win=64240 Len=0
345	19:43:31.885452	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=24136 Win=64240 Len=0
346	19:43:31.885519	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=25596 Win=64240 Len=0
347	19:43:31.885519	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=27056 Win=64240 Len=0
348	19:43:31.885553	13.91.16.71	192.168.223.128	TCP	60	443 → 49741 [ACK] Seq=2484 Ack=27857 Win=64240 Len=0

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

Ans: If we look at the frame section of the GET request we see that the time the packet arrived is **20:56:09.075255000**

9884	20:55:09.74214	117.259.240.8	192.168.223.128	HTTP	298	HTTP/1.1 206 Partial Content (application/x-chrome-extension...)
+ 10344	20:56:09.075255	192.168.223.128	23.60.169.18	HTTP	303	GET /c/msdownload/update/others/2021/09/35189963_096aa403570...
+ 10351	20:56:09.101035	23.60.169.18	192.168.223.128	HTTP	410	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)

```

▼ Frame 10344: 303 bytes on wire (2424 bits), 303 bytes captured (2424 bits) on interface \Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686}
  > Interface id: 0 (\Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 22, 2021 20:56:09.075255000 India Standard Time
      [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1632324369.075255000 seconds
      [Time delta from previous captured frame: 0.000365000 seconds]
      [Time delta from previous displayed frame: 179.351041000 seconds]
      [Time since reference or first frame: 581.419436000 seconds]
    Frame Number: 10344
    Frame Length: 303 bytes (2424 bits)
    Capture Length: 303 bytes (2424 bits)
  
```

The same section for the HTTP OK shows an arrival time of **20:56:09.101035000**

```

▼ Frame 10351: 410 bytes on wire (3280 bits), 410 bytes captured (3280 bits) on interface \Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686}
  > Interface id: 0 (\Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 22, 2021 20:56:09.101035000 India Standard Time
      [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1632324369.101035000 seconds
      [Time delta from previous captured frame: 0.000000000 seconds]
      [Time delta from previous displayed frame: 0.025780000 seconds]
      [Time since reference or first frame: 581.445216000 seconds]
    Frame Number: 10351
    Frame Length: 410 bytes (3280 bits)
    Capture Length: 410 bytes (3280 bits)
  
```

The difference of these 2 times gives :

$$0.101035000 - 0.075255000 = 0.02578$$

3. What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer? Support your answer with an appropriate screenshot from your computer.

If we look at the IP section of the GET request, the source and destination are shown.

The source is the local machine's address and the destination is the web server's public

My computer's address = 192.168.1.4

IP address 128.119.245.12 (gaia.cs.umass.edu)

Frame 8722: 555 bytes on wire (4440 bits), 555 bytes captured (4440 bits) on interface \Device\NPF_{044A503D-2D3A-4E83-A547-A0CF847228F7}, id 0
 > Ethernet II, Src: IntelCor_d8:b0:85 (e0:04:e8:d8:b0:85), Dst: GenexisI_59:ae:f8 (bc:62:d2:59:ae:f8)
 > Internet Protocol Version 4, Src: 192.168.1.4, Dst: 128.119.245.12
 > Transmission Control Protocol, Src Port: 60323, Dst Port: 80, Seq: 1, Ack: 1, Len: 501
 Hypertext Transfer Protocol
 > GET / HTTP/1.1\r\n
 Host: gaia.cs.umass.edu\r\n
 Connection: keep-alive\r\n
 Upgrade-Insecure-Requests: 1\r\n
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.82 Safari/537.36\r\n
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
 Referer: http://gaia.cs.umass.edu/\r\n
 Accept-Encoding: gzip, deflate\r\n
 Accept-Language: en-US,en;q=0.9\r\n
 Cookie: c2VwLW5vLXJ1ZGlyZWNB\r\n
 \r\n
 [Full request URI: http://gaia.cs.umass.edu/]
 [HTTP request 1/1]
 [Response in frame: 8728]

4. Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the “Selected Packet Only” and “Print as displayed” radial buttons, and then click OK

Ans: HTTP GET message

```
No. Time Source Destination Protocol Length Info
 10344 20:56:09.075255 192.168.223.128 23.60.169.18 HTTP 303 GET /c/msdownload/update/others/
2021/09/35189963_096aa403570075046a311dc3c0d575a43b48c3c0.cab HTTP/1.1
Frame 10344: 303 bytes on wire (2424 bits), 303 bytes captured (2424 bits) on interface \Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686}, id 0
  Interface id: 0 (\Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686})
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep 22, 2021 20:56:09.075255000 India Standard Time
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1632324369.075255000 seconds
  [Time delta from previous captured frame: 0.000365000 seconds]
  [Time delta from previous displayed frame: 179.351041000 seconds]
  [Time since reference or first frame: 581.419436000 seconds]
  Frame Number: 10344
  Frame Length: 303 bytes (2424 bits)
  Capture Length: 303 bytes (2424 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:http]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: VMware_72:57:6d (00:0c:29:72:57:6d), Dst: VMware_ec:5d:0f (00:50:56:ec:5d:0f)
Internet Protocol Version 4, Src: 192.168.223.128, Dst: 23.60.169.18
Transmission Control Protocol, Src Port: 64338, Dst Port: 80, Seq: 1, Ack: 1, Len: 249
Hypertext Transfer Protocol
  GET /c/msdownload/update/others/2021/09/35189963_096aa403570075046a311dc3c0d575a43b48c3c0.cab HTTP/1.1\r\n
  Connection: Keep-Alive\r\n
  Accept: */*\r\n
  User-Agent: Windows-Update-Agent/10.0.10011.16384 Client-Protocol/2.32\r\n
  Host: download.windowsupdate.com\r\n
  \r\n
  [Full request URI: http://download.windowsupdate.com/c/msdownload/update/others/
2021/09/35189963_096aa403570075046a311dc3c0d575a43b48c3c0.cab]
  [HTTP request 1/6]
  [Response in frame: 10351]
  [Next request in frame: 10353]
```

HTTP OK message

```
No.    Time          Source        Destination      Protocol Length Info
10351 20:56:09.101035  23.60.169.18    192.168.223.128  HTTP     410    HTTP/1.1 200 OK  (application/vnd.ms-cab-
compressed)
Frame 10351: 410 bytes on wire (3280 bits), 410 bytes captured (3280 bits) on interface \Device\NPF_{7299B546-D2B4-45D2-BC9A-
C701305A3686}, id 0
    Interface id: 0 (\Device\NPF_{7299B546-D2B4-45D2-BC9A-C701305A3686})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 22, 2021 20:56:09.101035000 India Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1632324369.101035000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.025780000 seconds]
    [Time since reference or first frame: 581.445216000 seconds]
    Frame Number: 10351
    Frame Length: 410 bytes (3280 bits)
    Capture Length: 410 bytes (3280 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:media]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: VMware_ec:5d:0f (00:50:56:ec:5d:0f), Dst: VMware_72:57:6d (00:0c:29:72:57:6d)
Internet Protocol Version 4, Src: 23.60.169.18, Dst: 192.168.223.128
Transmission Control Protocol, Src Port: 80, Dst Port: 64338, Seq: 7281, Ack: 250, Len: 356
[6 Reassembled TCP Segments (7636 bytes): #10346(1440), #10347(1460), #10348(1460), #10349(1460), #10350(1460), #10351(356)]
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Cache-Control: public,max-age=172800\r\n
    Content-Type: application/vnd.ms-cab-compressed\r\n
    Last-Modified: Wed, 22 Sep 2021 14:37:22 GMT\r\n
    Accept-Ranges: bytes\r\n
    ETag: "03d575abfafd71:0"\r\n
    Server: Microsoft-IIS/10.0\r\n
    X-Powered-By: ASP.NET\r\n
    Content-Length: 7281\r\n
    Date: Wed, 22 Sep 2021 15:26:08 GMT\r\n
    Connection: keep-alive\r\n
    X-CCC: IN\r\n
    X-CID: 2\r\n
    \r\n
    [HTTP response 1/6]
    [Time since request: 0.025780000 seconds]
    [Request in frame: 10344]
    [Next request in frame: 10353]
    [Next response in frame: 10360]
    [Request URI: http://download.windowsupdate.com/c/msdownload/update/others/
2021/09/35189715_f2fa2f2c55b978a7bcc377bda307a3b8aa14519.cab]
    File Data: 7281 bytes
    Media Type
```

5. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?

Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request? What is the status code and phrase in the response?

Ans:

- My browser sent 3 HTTP GET request to the server. The Packet that contained the GET message was packet number 5350, 5359, 5369.
- The packet that contains the status code and phrase which the server sent in response to the GET message was packet number 5352, 5361 and 5371.
- The code and phrase in the response was:

➤ HTTP317 HTTP/1.1 304 Not Modified

➤ HTTP322 HTTP/1.1 304 Not Modified

➤ HTTP377 HTTP/1.1 304 Not Modified

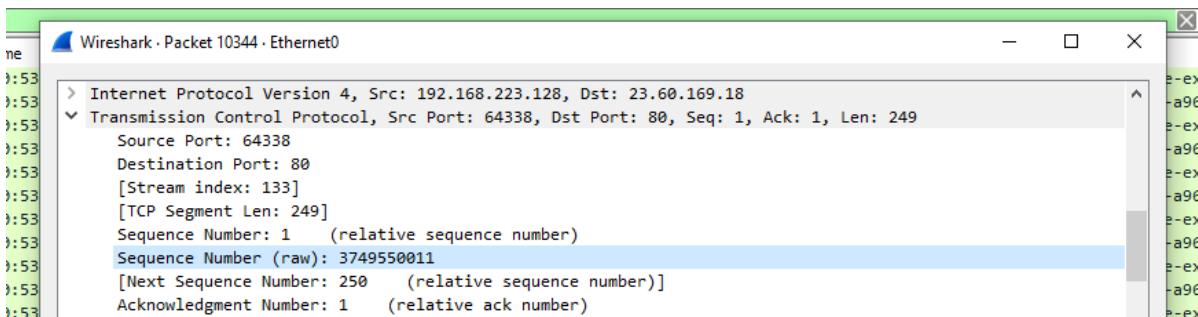
5350	213.763729	192.168.1.4	104.114.101.136	HTTP	281 GET / HTTP/1.1
5352	213.780603	104.114.101.136	192.168.1.4	HTTP	317 HTTP/1.1 304 Not Modified
5359	213.808749	192.168.1.4	23.60.169.19	HTTP	307 GET /DSTROOTCAX3CRL.crl HTTP/1.1
5361	213.821499	23.60.169.19	192.168.1.4	HTTP	322 HTTP/1.1 304 Not Modified
5369	213.868581	192.168.1.4	151.101.158.133	HTTP	265 GET /root-r2.crl HTTP/1.1
5371	213.890802	151.101.158.133	192.168.1.4	HTTP	377 HTTP/1.1 304 Not Modified

> Frame 5350: 281 bytes on wire (2248 bits), 281 bytes captured (2248 bits) on interface \Device\NPF_{044A503D-2D3A-4E83-A547-A0CF847228F7}, id 0
 > Ethernet II, Src: IntelCor_d8:b0:85 (e0:d4:e8:d8:b0:85), Dst: GenexisI_59:ae:f8 (bc:62:d2:59:ae:f8)
 > Internet Protocol Version 4, Src: 192.168.1.4, Dst: 104.114.101.136
 > Transmission Control Protocol, Src Port: 58672, Dst Port: 80, Seq: 1, Ack: 1, Len: 227
 ▾ Hypertext Transfer Protocol
 > GET / HTTP/1.1\r\n

6. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

Is there any HTTP header information in the transmitted data associated with TCP segmentation? For this question you may want to think about at what layer each protocol operates, and how the protocols at the different layers interoperate.

Ans: The data was sent in 1 TCP segments to the browser



TCP Segment Information

>	Internet Protocol Version 4, Src: 192.168.223.128, Dst: 23.60.169.18
	▀ Transmission Control Protocol, Src Port: 64338, Dst Port: 80, Seq: 1, Ack: 1, Len: 249
	Source Port: 64338
	Destination Port: 80
	[Stream index: 133]
	[TCP Segment Len: 249]
	Sequence Number: 1 (relative sequence number)
	Sequence Number (raw): 3749550011
	[Next Sequence Number: 250 (relative sequence number)]
	Acknowledgment Number: 1 (relative ack number)
	Acknowledgment number (raw): 1813868173
	0101 = Header Length: 20 bytes (5)

EXPERIMENT NO:24

FAMILIARISATION TO HYPERVISORS AND VIRTUALMACHINES

Virtual Machine

- A virtual machine is a virtual representation, or emulation, of a physical computer. They are often referred to as a guest while the physical machine they run on is referred to as the host.
- Virtualization makes it possible to create multiple virtual machines, each with their own operating system (OS) and applications, on a single physical machine. A VM cannot interact directly with a physical computer. Instead, it needs a lightweight software layer called a hypervisor to coordinate between it and the underlying physical hardware.

HYPervisor

- Hypervisor is a software program that manages multiple operating systems (or multiple instances of the same operating system) on a single computer system.
- The hypervisor manages the system's processor, memory, and other resources to allocate what each operating system requires.
- Hypervisors are designed for a particular processor architecture and may also be called virtualization managers.

Hypervisor Types

Type 1: native (bare-metal) hypervisors

The Hypervisor runs directly on the host's hardware to control the hardware and to manage guest operating systems.

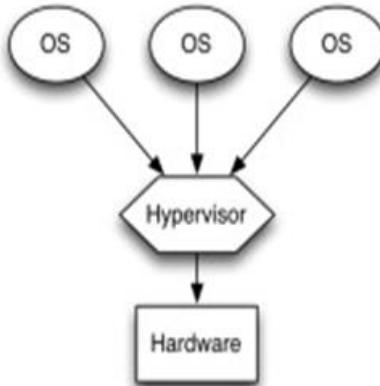
E.g., Xen, VMWare ESXi, Microsoft Hyper-V

Type 2: hosted hypervisors

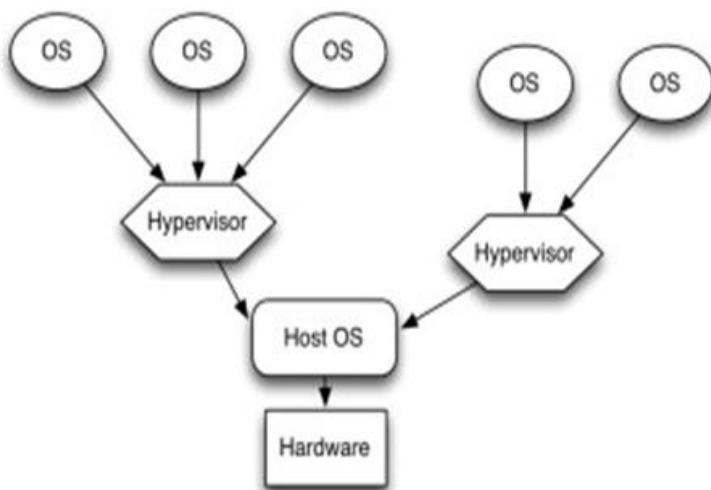
These hypervisors run on a conventional operating system just as other computer programs do.

Eg. VMWare Workstation, VirtualBox

HYPERVERISOR TYPES



Type 1 Hypervisor



Type 2 Hypervisor

Benefits of hypervisor

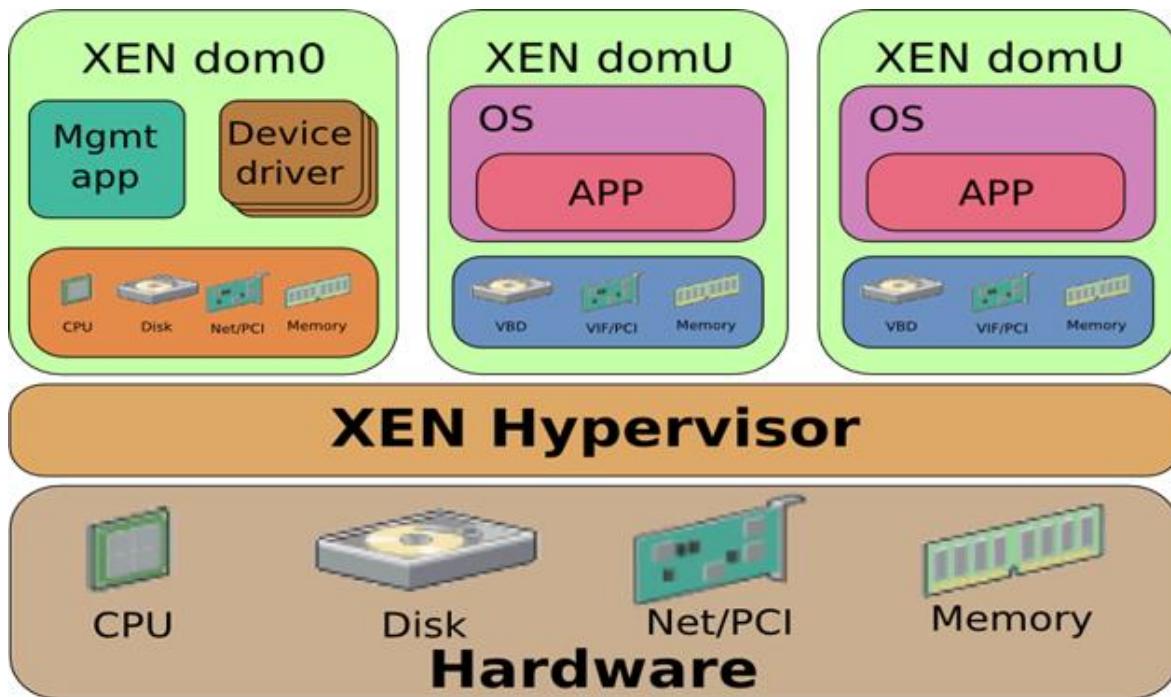
- Speed
- Efficiency
- Flexibility
- Portability

Popular Hypervisors

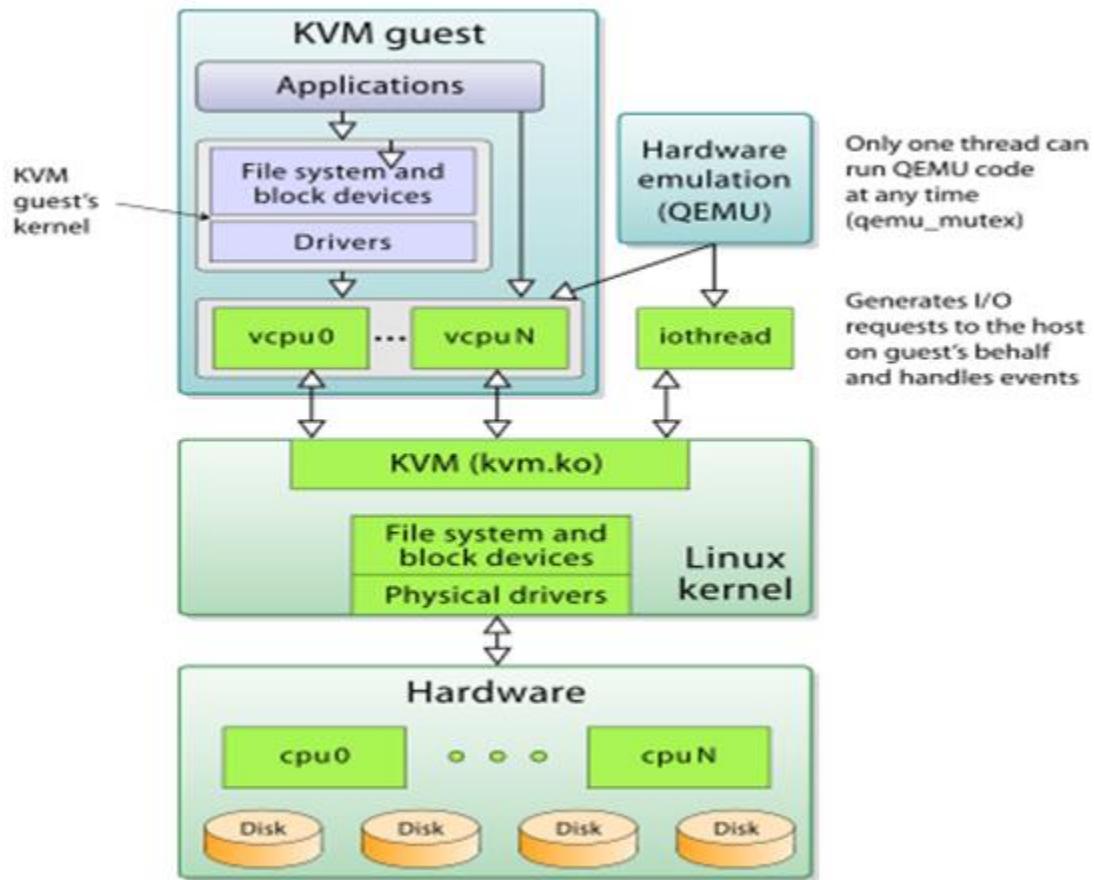
- Xen
- KVM(Kernel Based VM)
- SAN
- NAS
- Xen

EXPERIMENT NO: 25**FAMILIARISATION TO XEN OR KVM****XEN**

“Xen is an open-source paravirtualization technology that provides a platform for running multiple operating systems in parallel on one physical hardware resource”
 Originally developed in 2003 at the University of Cambridge Computer Laboratory.

**KVM****Kernel-based Virtual Machine (KVM)**

- Kernel-based Virtual Machine (KVM) is an open source virtualization technology built into Linux. Specifically, KVM lets you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines (VMs).
- KVM converts Linux into a type-1 (bare-metal) hypervisor. All hypervisors need some operating system-level components—such as a memory manager, process scheduler, input/output (I/O) stack, device drivers, security manager, a network stack, and more—to run VMs. KVM has all these components because it's part of the Linux kernel. Every VM is implemented as a regular Linux process, scheduled by the standard Linux scheduler, with dedicated virtual hardware like a network card, graphics adapter, CPU(s), memory, and disks.



DOCKER

Docker overview

- Docker is an open platform for developing, shipping, and running applications.
- Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
- With Docker, you can manage your infrastructure in the same ways you manage your applications.
- By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

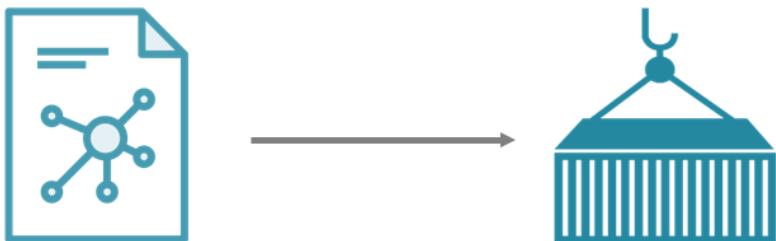
What is a container?

- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
- The isolation and security allow you to run many containers simultaneously on a given host.
- Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host.
- You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

What is a Docker image?

- A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template.
- Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments.

The Role of Images and Containers



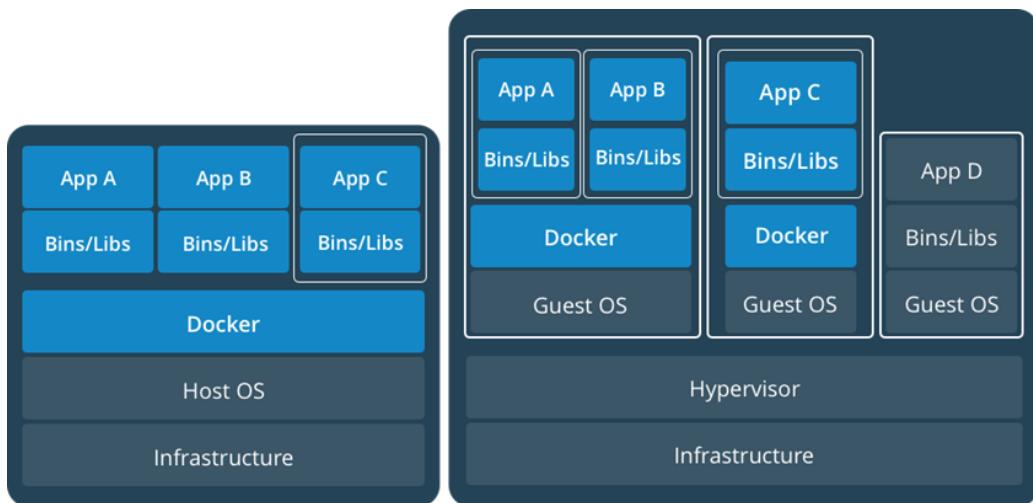
Docker Image

Docker Container

Example: Ubuntu with Node.js and Application Code

Created by using an image. Runs your application.

Docker is better...



Docker installation

---->\$ sudo apt install docker.io

```

robin@robin-VirtualBox: $ sudo apt install docker.io
[sudo] password for robin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 78.7 MB of archives.
After this operation, 386 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu groovy/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 bridge-utils amd64 1.6-3ubuntu1 [30.9 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 runc amd64 1.0.0-rc95-0ubuntu1~20.10.1 [4,075 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu groovy-updates/main amd64 containerd amd64 1.5.2-0ubuntu1~20.10.2 [31.7 MB]
21% [4 containerd 10.2 MB/31.7 MB 32%] 1,141 kB/s 56s

```

Version check

----->\$ docker --version

```

robin@robin-VirtualBox: $ docker --version
Docker version 20.10.2, build 20.10.2-0ubuntu1~20.10.1
robin@robin-VirtualBox: ~

```

- Check whether it is running or not

\$ sudo systemctl status docker

```

robin@robin-VirtualBox:~$ sudo systemctl status docker
[sudo] password for robin:
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2021-09-20 20:13:30 IST; 19min ago
    TriggeredBy: ● docker.socket
      Docs: https://docs.docker.com
   Main PID: 1000 (dockerd)
     Tasks: 10
    Memory: 69.4M
      CGroup: /system.slice/docker.service
              └─1000 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 20 20:13:22 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:22.76974>
Sep 20 20:13:22 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:22.77002>
Sep 20 20:13:22 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:22.77037>
Sep 20 20:13:22 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:22.77147>
Sep 20 20:13:27 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:27.41679>
Sep 20 20:13:27 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:27.56655>
Sep 20 20:13:30 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:30.12663>
Sep 20 20:13:30 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:30.15959>
Sep 20 20:13:30 robin-VirtualBox systemd[1]: Started Docker Application Contain>
Sep 20 20:13:30 robin-VirtualBox dockerd[1000]: time="2021-09-20T20:13:30.62695>
[lines 1-21/21 (END)]

```

- **If not active**

\$ sudo systemctl enable –now docker

- **List all the images you have locally**

\$ sudo docker images

```

robin@robin-VirtualBox:/ $ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
python latest a5210955ee89 2 weeks ago 911MB
robin@robin-VirtualBox:/ $

```

- **Pull an image from the Docker registry**

\$ sudo docker pull <image_name>:<tag>

- **Run a docker**

\$ sudo docker run <image_name>:<tag>

```
hari@hari-Omen15:~$ sudo docker run python:latest
[sudo] password for hari:
Unable to find image 'python:latest' locally
latest: Pulling from library/python
955615a668ce: Pull complete
2750ef5f69a5: Pull complete
911ea9f2bd51: Pull complete
27b0a22ee906: Pull complete
8584d51a9262: Pull complete
524774b7d363: Pull complete
af193b9b3d11: Pull complete
aacb0e56e8f3: Pull complete
46cd7abc9e93: Pull complete
Digest: sha256:e6654fa815122b13242fc9ff513e2d14b00548ba6eaf4d3b03f2f261d85272d
Status: Downloaded newer image for python:latest
hari@hari-Omen15:~$ sudo docker run python:latest
hari@hari-Omen15:~$
```

- List all the running containers

```
$ sudo docker ps -a
```

```
robin@robin-VirtualBox:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
24cf0335e079        python:latest      "python3"          4 minutes ago     Exited (0) About a minute ago
robin@robin-VirtualBox:~$
```

- Remove a container

```
$ sudo docker rm <container id>
```

```
robin@robin-VirtualBox:~$ sudo docker rm 24cf0335e079
24cf0335e079
robin@robin-VirtualBox:~$
```

EXPERIMENT NO: 26

INSTALLATION OF SOFTWARE FROM SOURCE CODE

- Source code software must be compiled and installed.
- Usually comes in a compressed archive, called a tarball with .tar or .tar.gz ending.
- Archive includes source, configure script, makefile and install scripts.

Package Managers

- Automate the installation, removal, and management of the software applications.
- Only track software installed using the package manager.
- Similar to Add/Remove programs control panel in MS Windows

Configure Script:

Inspects system for requirements and configures the “makefile” .

Make:

- Automates the compilation of programming source code for the target system.
- “makefile” defines the necessary steps to build the application.
- They are far from perfect
- There is no central database to track applications installed with make.
- Removal of applications may or may not be supported by the make file.
- “makefile” contains installation parameters, variables, and setup instructions.
- “make” and “make install” commands are run to compile and install software.
- There is no central database to track applications installed with make.
- Removal of applications may or may not be supported by the make file.
- “makefile” contains installation parameters, variables, and setup instructions.
- “make” and “make install” commands are run to compile and install software.

Make Command:

- Source code distributed as “gzipped tarballs”.
- After unpacking the code you must check the README file for specific install instructions.

```
$ configure  
$ make  
$ make install
```

Installation Steps:

Step 1: Open the Linux terminal and enter

```
sudo apt update
```

```
robin@robin-VirtualBox:~$ sudo apt update
[sudo] password for robin:
Hit:1 http://in.archive.ubuntu.com/ubuntu groovy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu groovy-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu groovy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu groovy-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
robin@robin-VirtualBox:~$
```

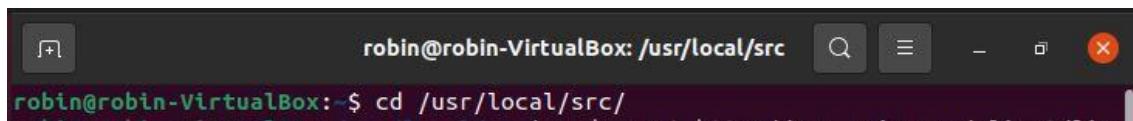
Step 2: Enter

```
sudo apt install build-essential
```

```
robin@robin-VirtualBox:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++
  g++-10 gcc gcc-10 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin
  libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libfakeroot libgcc-10-dev
  libitm1 liblsan0 libnsl-dev libquadmath0 libstdc++-10-dev libtirpc-dev
  libtsan0 libubsan1 linux-libc-dev make manpages-dev rpcsvc-proto
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-10-multilib gcc-10-doc
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-10-multilib
  gcc-10-locales glibc-doc libstdc++-10-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-10 gcc gcc-10 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libfakeroot libgcc-10-dev libitm1 liblsan0 libnsl-dev libquadmath0
  libstdc++-10-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev make
  manpages-dev rpcsvc-proto
0 upgraded, 35 newly installed, 0 to remove and 0 not upgraded.
Need to get 46.8 MB of archives.
After this operation, 185 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu groovy/main amd64 binutils-common amd
64 2.35.1-1ubuntu1 [212 kB]
```

Step 3: Enter

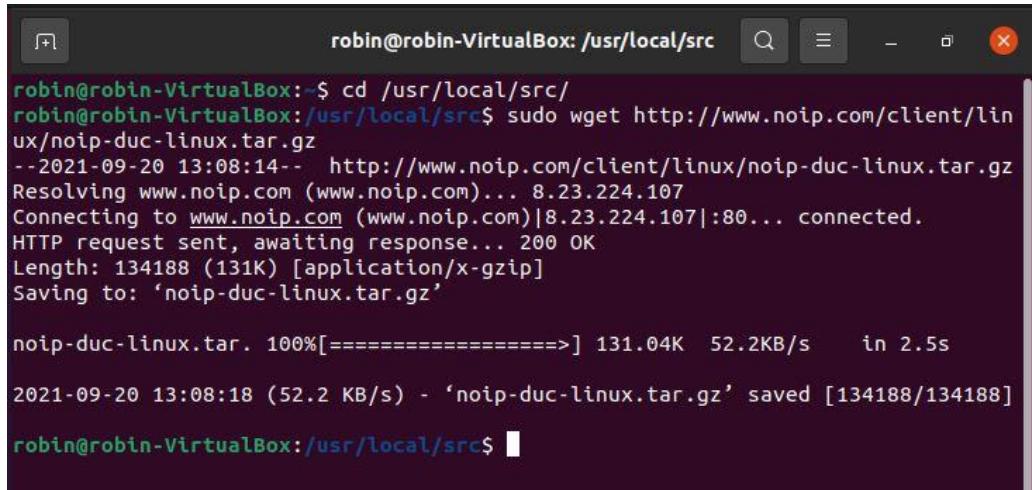
```
cd /usr/local/src/
```



```
robin@robin-VirtualBox:~$ cd /usr/local/src
```

Step 4: Enter

```
sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
```



```
robin@robin-VirtualBox:~$ cd /usr/local/src
robin@robin-VirtualBox:/usr/local/src$ sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
--2021-09-20 13:08:14--  http://www.noip.com/client/linux/noip-duc-linux.tar.gz
Resolving www.noip.com (www.noip.com)... 8.23.224.107
Connecting to www.noip.com (www.noip.com)|8.23.224.107|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134188 (131K) [application/x-gzip]
Saving to: 'noip-duc-linux.tar.gz'

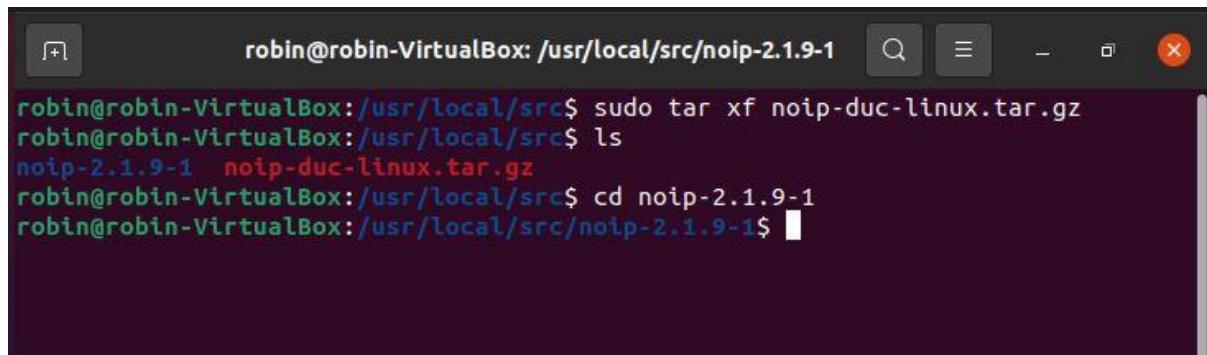
noip-duc-linux.tar.gz 100%[=====] 131.04K 52.2KB/s   in 2.5s

2021-09-20 13:08:18 (52.2 KB/s) - 'noip-duc-linux.tar.gz' saved [134188/134188]

robin@robin-VirtualBox:/usr/local/src$
```

Step 5: Enter

```
sudo tar xf noip-duc-linux.tar.gz and cd noip-2.1.9-1/
```



```
robin@robin-VirtualBox:/usr/local/src$ sudo tar xf noip-duc-linux.tar.gz
robin@robin-VirtualBox:/usr/local/src$ ls
noip-2.1.9-1  noip-duc-linux.tar.gz
robin@robin-VirtualBox:/usr/local/src$ cd noip-2.1.9-1
robin@robin-VirtualBox:/usr/local/src/noip-2.1.9-1$
```

Step 6: Enter

```
sudo make install
```

```
robin@robin-VirtualBox:/usr/local/src/noip-2.1.9-1$ sudo make install
gcc -Wall -g -Dlinux -DPREFIX=\"/usr/local\" noip2.c -o noip2
noip2.c: In function 'dynamic_update':
noip2.c:1595:6: warning: variable 'i' set but not used [-Wunused-but-set-variable]
  1595 |     int i, x, is_group, retval, response;
          ^
noip2.c: In function 'domains':
noip2.c:1826:13: warning: variable 'x' set but not used [-Wunused-but-set-variable]
  1826 |         int x;
          ^
noip2.c: In function 'hosts':
noip2.c:1838:20: warning: variable 'y' set but not used [-Wunused-but-set-variable]
  1838 |             int     x, y, z;
          ^
noip2.c: In function 'autoconf':
noip2.c:2538:19: warning: '%s' directive writing up to 255 bytes into a region
of size 247 [-Wformat-overflow=]
  2538 |     sprintf(line, "%s%s%s%s", USTRNG, login, PWDSTRNG, password);
          ^
noip2.c:2538:2: note: 'sprintf' output between 16 and 526 bytes into a destination
of size 256
  2538 |     sprintf(line, "%s%s%s%s", USTRNG, login, PWDSTRNG, password);
          ^
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
```

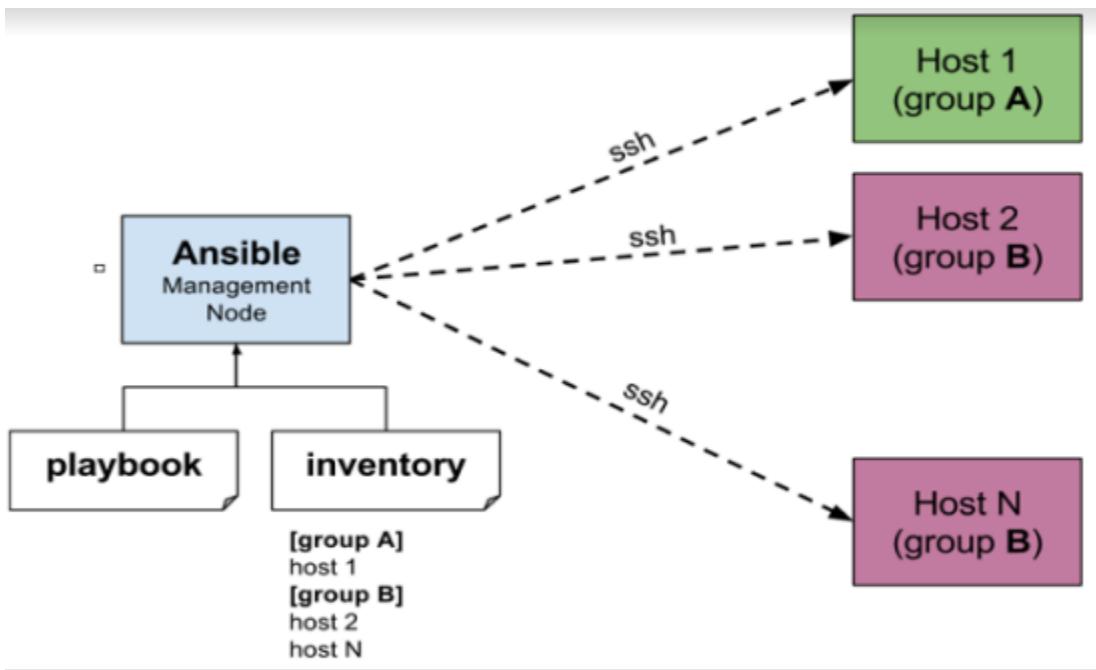
EXPERIMENT NO: 27
DEPLOY LINUX VM USING ANSIBLE PLAYBOOK

What is Ansible ?

- Ansible is simple open source IT engine which automates application deployment, intra service orchestration, cloud provisioning and many other IT tools.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure.
- Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML (It's a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored)which is very easy for humans to understand, read and write. Hence the advantage is that even the IT infrastructure support guys can read and understand the playbook and debug if needed (YAML – It is in human readable form).
- Ansible is designed for multi-tier deployment. Ansible does not manage one system at time, it models IT infrastructure by describing all of your systems are interrelated. Ansible is completely agentless which means Ansible works by connecting your nodes through ssh(by default). But if you want other method for connection like Kerberos, Ansible gives that option to you.

How do Ansible playbooks work?

- Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. Ansible then executes these modules (over SSH by default), and removes them when finished. Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.



Installation Process

- Ansible can be run from any machine with Python 2 (versions 2.6 or 2.7) or Python 3 (versions 3.5 and higher) installed.
- Ansible can be installed on control machine which have above mentioned requirements in different ways. You can install the latest release through Apt, yum, pkg, pip, OpenCSW, pacman, etc.

Installation through Apt on Ubuntu Machine

- For installing Ansible you have to configure PPA on your machine. For this, you have to run the following line of code:

```
sudo apt-add-repository ppa:ansible/ansible
```

```

robin@robin-VirtualBox:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb http://ppa.launchpad.net/ansible/ansible/ubuntu/ groovy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and system's easier to deploy. Avoid writing scripts or custom code to deploy and update your applications—automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-groovy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-groovy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://in.archive.ubuntu.com/ubuntu groovy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu groovy-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu groovy-backports InRelease
Ign:4 http://ppa.launchpad.net/ansible/ansible/ubuntu groovy InRelease
Hit:5 http://security.ubuntu.com/ubuntu groovy-security InRelease
Err:6 http://ppa.launchpad.net/ansible/ansible/ubuntu groovy Release
  404  Not Found [IP: 91.189.95.85 80]

```

sudo apt update

```

robin@robin-VirtualBox:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu groovy InRelease
Ign:2 http://ppa.launchpad.net/ansible/ansible/ubuntu groovy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu groovy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu groovy-backports InRelease
Err:5 http://ppa.launchpad.net/ansible/ansible/ubuntu groovy Release
  404  Not Found [IP: 91.189.95.85 80]
Hit:6 http://security.ubuntu.com/ubuntu groovy-security InRelease
Reading package lists... Done
E: The repository 'http://ppa.launchpad.net/ansible/ansible/ubuntu groovy Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
robin@robin-VirtualBox:~$ 

```

sudo apt install ansible

```

robin@robin-VirtualBox:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-distutils python3-dnspython
  python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth python3-pycryptodome
  python3-requests-kerberos python3-requests-ntlm
  python3-selinux python3-winrm python3-xmldict
Suggested packages:
  cowsay sshpass python-jinja2-doc ipython3 python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-crypto python3-distutils python3-dnspython
  python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth python3-pycryptodome
  python3-requests-kerberos python3-requests-ntlm
  python3-selinux python3-winrm python3-xmldict
0 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 19.9 MB of archives.

```

Setting up Inventory File

```
GNU nano 4.8                               /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group
[ Read 44 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^  Go To Line
```

ansible-inventory --list -y

Testing Connection

- After setting up the inventory file to include your servers, it's time to check if Ansible is able to connect to these servers and run commands via SSH.
- You can use the -u argument to specify the remote system user. When not provided, Ansible will try to connect as your current system user on the control node.
- ansible all -m ping -u root

PaaS (Platform-as-a-Service)

- Platform-as-a-service (PaaS) is a model of cloud service delivery where a third-party cloud service provider delivers some hardware and software tools, often those needed for application hosting or development, to customers over the internet. The key benefit of the PaaS model is that it enables users to access hardware and software that can be used to develop and run applications without having to purchase, install and maintain the infrastructure.
- Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.

What is Microsoft Azure used for?

- Microsoft Azure consists of numerous service offerings, its use cases are extremely diverse. Running virtual machines or containers in the cloud is one of the most popular uses for Microsoft Azure. These compute resources can host infrastructure components, such as domain name system (DNS) servers; Windows Server services -- such as Internet Information Services (IIS); or third-party applications. Microsoft also supports the use of third-party operating systems, such as Linux.

AZURE SERVICES

- ▶ **Compute.** These services enable a user to deploy and manage VMs, containers and batch jobs, as well as support remote application access. Compute resources created within the Azure cloud can be configured with either public IP addresses or private IP addresses, depending on whether the resource needs to be accessible to the outside world.
- ▶ **Mobile.** These products help developers build cloud applications for mobile devices, providing notification services, support for back-end tasks, tools for building application program interfaces (APIs) and the ability to couple geospatial context with data.
- ▶ **Web.** These services support the development and deployment of web applications. They also offer features for search, content delivery, API management, notification and reporting.
- ▶ **Storage.** This category of services provides scalable cloud storage for structured and unstructured data. It also supports big data projects, persistent storage and archival storage.
- ▶ **Analytics.** These services provide distributed analytics and storage, as well as features for real-time analytics, big data analytics, data lakes, machine learning (ML), business intelligence (BI), internet of things (IoT) data streams and data warehousing.
- ▶ **Networking.** This group includes virtual networks, dedicated connections and gateways, as well as services for traffic management and diagnostics, load balancing, DNS hosting and network protection against distributed denial-of-service (DDoS) attacks.
- ▶ **Media and content delivery network (CDN).** These CDN services include on-demand streaming, digital rights protection, encoding and media playback and indexing.
- ▶ **Integration.** These are services for server backup, site recovery and connecting private and public clouds.
- ▶ **Identity.** These offerings ensure only authorized users can access Azure services and help protect encryption keys and other sensitive information in the cloud. Services include support for Azure Active Directory and multifactor authentication (MFA).
- ▶ **Internet of things.** These services help users capture, monitor and analyze IoT data from sensors and other devices. Services include notifications, analytics, monitoring and support for coding and execution.
- ▶ **DevOps.** This group provides project and collaboration tools, such as Azure DevOps -- formerly Visual

Studio Team Services -- that facilitate DevOps software development processes. It also offers features for application diagnostics, DevOps tool integrations and test labs for build tests and experimentation.

- ▶ **Development.** These services help application developers share code, test applications and track potential issues. Azure supports a range of application programming languages, including JavaScript, Python, .NET and Node.js. Tools in this category also include support for Azure DevOps, software development kits (SDKs) and blockchain.
- ▶ **Security.** These products provide capabilities to identify and respond to cloud security threats, as well as manage encryption keys and other sensitive assets.
- ▶ **Artificial intelligence (AI) and machine learning.** This is a wide range of services that a developer can use to infuse artificial intelligence, machine learning and cognitive computing capabilities into applications and data sets.
- ▶ **Containers.** These services help an enterprise create, register, orchestrate and manage huge volumes of containers in the Azure cloud, using common platforms such as Docker and Kubernetes.
- ▶ **Databases.** This category includes Database as a Service (DBaaS) offerings for SQL and NoSQL, as well as other database instances -- such as Azure Cosmos DB and Azure Database for PostgreSQL. It also includes Azure SQL Data Warehouse support, caching and hybrid database integration and migration features. Azure SQL is the platform's flagship database service. It is a relational database that provides SQL functionality without the need for deploying a SQL server.
- ▶ **Migration.** This suite of tools helps an organization estimate workload Migration costs and perform the actual migration of workloads from local data centers to the Azure cloud.
- ▶ **Management and governance.** These services provide a range of backup, recovery, compliance, automation, scheduling and monitoring tools that can help a cloud administrator manage an Azure deployment.
- ▶ **Mixed reality.** These services are designed to help developers create content for the Windows Mixed Reality environment.
- ▶ **Blockchain.** The Azure Blockchain Service allows you to join a blockchain consortium or to create your own.

HOME PAGE OF AZURE

VM DEPLOYMENT WITHOUT ANSIBLE

1. Head to a virtual machine from the Azure homescreen.

2. Click on **create->virtual machine**.

3. Fill the details that you need to create a virtual machine and hit **Review+create**.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. On the left, there's a sidebar titled 'Virtual machines' with a 'Create' button and a 'No virtual machines to display' message. The main area is titled 'Create a virtual machine' and has tabs for 'Basics', 'Disks', 'Networking', 'Management', 'Advanced', 'Tags', and 'Review + create'. The 'Basics' tab is active. It contains instructions to create a virtual machine running Linux or Windows from the marketplace or a customized image. Below this, 'Project details' allow selecting a subscription (set to 'Azure for Students') and a resource group ('(New) Resource group' with a 'Create new' link). 'Instance details' include fields for 'Virtual machine name' (empty) and 'Region' (set to '(US) East US'). At the bottom, there are 'Review + create' and 'Next : Disks >' buttons.

USING ANSIBLE

To Configure LINUX VM Using Ansible Playbook ,

- First we need an azure account, to get a free account:

-> <https://azure.microsoft.com/en-in/free/>

- Create a resource group
- Create a virtual network
- Create a public IP address
- Create a network security group
- Create a virtual network interface card
- Create a virtual machine

Open The Azure Shell, we can directly run Ansible in Azure Cloud Shell, where Ansible is **pre-installed**.

In case not installed,

#Update all packages that have available updates.

```
sudo yum update -y
```

Install Python 3 and pip.

```
sudo yum install -y python3-pip
```

Upgrade pip3.

```
sudo pip3 install --upgrade pip
```

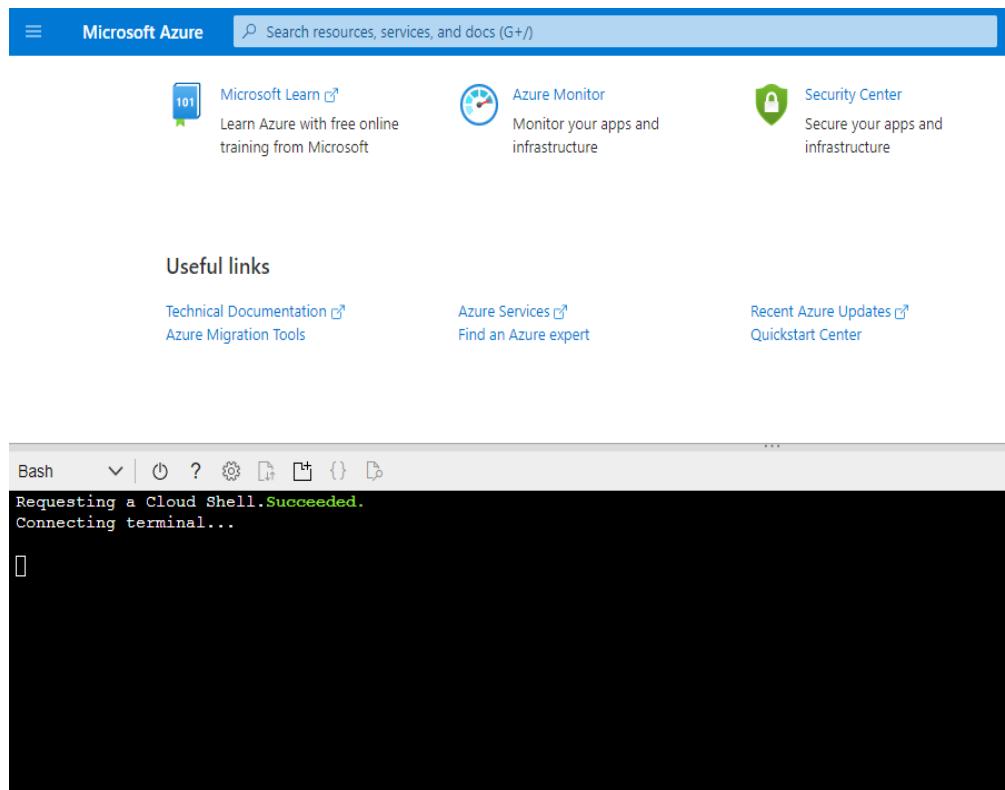
Install Ansible. pip3 install "ansible==2.9.17"

Install Ansible azure_rm module for interacting with Azure.

```
pip3 install ansible[azure]
```

To check the ansible version installed,

```
ansible --version
```



Creating a Linux vm(using ansible)

- ▶ name: Create Linux VM

```

hosts: localhost
connection: local
tasks:
  - name: Create resource group to hold VM
    azure_rm_resourcegroup:
      name: ansible-test-rg-vm
      location: eastus

  ▶ name: Create virtual network
    azure_rm_virtualnetwork:
      resource_group: ansible-test-rg-vm
      name: myVnet
      address_prefixes: "10.0.0.0/16"

  ▶   - name: Add subnet
    azure_rm_subnet:
      resource_group: ansible-test-rg-vm
      name: mySubnet
      address_prefix: "10.0.1.0/24"
      virtual_network: myVnet

  ▶   - name: Create public IP address
    azure_rm_publicipaddress:
      resource_group: ansible-test-rg-vm
      allocation_method: Static
      name: myPublicIP
      register: output_ip_address

  - name: Dump public IP for VM which will be create debug:
    msg: "The public IP is {{ output_ip_address.state.ip_address }}."

```

- ▶ - name: Create Network Security Group that allows SSH
 - azure_rm_securitygroup:
 - resource_group: ansible-test-rg-vm
 - name: myNetworkSecurityGroup
 - rules:
 - name: SSH
 - protocol: Tcp
 - destination_port_range: 22
 - access: Allow
 - priority: 1001
 - direction: Inbound
- ▶ name: Create virtual network interface card
 - azure_rm_networkinterface:
 - resource_group: ansible-test-rg-vm
 - name: myNIC
 - virtual_network: myVnet
 - subnet: mySubnet
 - public_ip_name: myPublicIP
 - security_group: myNetworkSecurityGroup
- ▶ - name: Create VM
 - azure_rm_virtualmachine:
 - resource_group: ansible-test-rg-vm
 - name: Linuxvmm
 - vm_size: Standard_DS1_v2
 - admin_username: azureuser
 - admin_password: password(password must contain numeric, special, uppercase, lowercase also

password must be between 12-72 characters long)

ssh_password_enabled: true

network_interfaces: myNIC



image:

offer: CentOS

publisher: OpenLogic

sku: '7.5'

version: latest

► To access the VM created ,open a shell and type:

ssh username@ public ip address of vm