

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



**20MCA136 – NETWORKING & SYSTEM ADMINISTRATION
LAB
PRACTICAL RECORD BOOK
Second Semester MCA 2022**

Submitted by:

NAME: BINY BIJU

REG NO: TKM21MCA-2013

DEPARTMENT OF COMPUTER APPLICATION
TKM COLLEGE OF ENGINEERING KOLLAM – 691005



Certificate

This is a bonafide record of the work done by BINCY BIJU in the Second Semester in NETWORKING & SYSTEM ADMINISTRATION LAB Course (20MCA136) towards the partial fulfilment of the degree of Master of Computer Applications during the academic year 2022.

Staff Member in-charge

Examiner

INDEX

SL.NO	PROGRAMS	PAGE.NO
	Cycle 1	
1	Familiarisation to computer hardware	1
2	Specification of desktop and server computers	4
3	Installation of linux on virtual box in windows 10	7
4	Installation of windows on virtual box in ubuntu	13
5	Installation of windows operating system on a bare machine	18
6	Installation of linux over windows operating system	21
7	Installation of server system	23
	Cycle 2	
8	Study on command line text editors	26
9	Basic Linux Commands	28
10	Familiarisation with Vi Editor	38
11	Familiarisation with EMACS editor	41
12	Linux File System	45
	Cycle 3	
13	Familiarisation to linux shell and shell scripting	54
14	Linux shell scripting problems	58
	Cycle 4	
15	Installation of LAMP	65
16	Installation of Laravel	69
	Cycle 5	
17	Familiarisation to command Line tool for networking	72

18	Familiarisation of Static and Dynamic IP	79
19	Concept of subnets and CIDR address scheme	82
20	Concept of subnet mask	85
21	Setting up of firewall on LAN	87
22	Wireshark and TCP Dump	88
23	Analyse Packets using wireshark	94
	Cycle 6	
24	Familiarisation to hypervisors and Virtual machines	97
25	Familiarisation to Xen or KVM	99
26	Installation of software from source code	102
27	Deploy Linux virtual machine using Ansible Playbook	105

CYCLE 1

EXPERIMENT 1

FAMILARIZATION OF COMPUTER HARDWARE

MOTHERBOARD

The motherboard serves as a single platform to connect all of the parts of a computer together. It connects the CPU, memory, hard drives, optical drives, video card, sound card, and other ports and expansion cards directly or via cables. It can be considered as the backbone of a computer. Unlike a backplane, a motherboard usually contains significant sub-systems, such as the central processor, the chipset's input/output and memory controllers, interface connectors, and other components integrated for general use.

Intel® Desktop Board D945GCPE



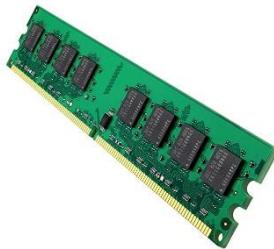
Specifications:

Board Form:	FactormicroATX7
Socket:	LGA775
Board Chipset:	Intel® 82945GC Graphics and Memory Controller
Max Memory Size (dependent on memory type):	2 GB
Memory Types:	DDR2 533/667
USB 2.0 Configuration (External + Internal):	4+4
Audio chip:	Realtek ALC662
Motherboard form factor:	micro ATX
Power source type:	ATX
Width:	218.4 mm
Depth:	243.8 mm

RAM MODULES

A narrow-printed circuit board that holds memory chips (RAM chips). The common architecture for desktop computers is the dual in-line memory module (DIMM), which transfers 64 bits at a time. Because of space limitations, laptops use small outline DIMMs

(SODIMMs). The modules are keyed with notches in different places so they cannot be inserted into the wrong slots. RAM allows your computer to perform many of its everyday tasks, such as loading applications, browsing the internet, editing a spreadsheet, or experiencing the latest game. Memory also allows you to switch quickly among these tasks, remembering where you are in one task when you switch to another task. As a rule, the more memory you have, the better.



Specifications:

Manufacturer:	Hynix
Manufacturer Part:	HYMD564M646CP6-CF
Capacity:	2GB
Memory Technology:	DDR2
Computer Memory Type:	DDR2 SDRAM
Data Transfer Rate:	667Mhz

INTERNAL STORAGE DEVICES

Most computers have some form of internal storage. The most common type of internal storage is the hard disk. At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices. It will also be used to store the applications software that you use and more than likely, the original copies of your data files.

Internal storage allows the data and applications to be loaded very rapidly into memory, ready for use. The data can be accessed much faster than data which is stored on an external storage device. This is because internal storage devices are connected directly to the motherboard and its data bus whereas external devices are connected through a hardware interface such as USB, which means they are considerably slower to access.



Specifications:

Manufacturer:	SEAGATE
Size:	500 GB

Connector type:	SATA
Data transfer rate:	3.0 Gbps
Storage size:	500GB
Rotation Speed:	7200RPM
Type:	HDD

SWITCHED-MODE POWER SUPPLY (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.



Specifications:

Voltage:	220V
Current:	3A/5A
Frequency:	50/60 Hz

INTERFACING PORTS

In computer hardware, a port serves as an interface between the computer and other computers or peripheral devices. In computer terms, a port generally refers to the part of a computing device available for connection to peripherals such as input and output devices. Computer ports have many uses, to connect a monitor, webcam, speakers, or other peripheral devices. On the physical layer, a computer port is a specialized outlet on a piece of equipment to which a plug or cable connects. Electronically, the several conductors where the port and cable contacts connect, provide a method to transfer signals between devices.

VGA (VIDEO GRAPHICS ARRAY)

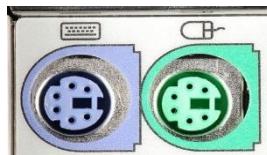
Video Graphics Array is a standard type of connection for video devices such as monitors and projectors. Generally, it refers to the types of cables, ports, and connectors used to connect monitors to video cards. While this technology is still in use today, it's rapidly being replaced by newer interfaces like DVI and HDMI. The Video Graphics Array (VGA) connector is a

standard connector used for computer video output. Originating with the 1987 IBM PS/2 and its VGA graphics system, the 15-pin connector went on to become ubiquitous on PCs, as well as many monitors, projectors and high-definition television sets.



PS/2 PORTS

The PS/2 (Personal System/2) port, also referred to as the mouse port or keyboard port, was developed by IBM. It is used to connect a computer mouse or keyboard to an IBM compatible computer. The PS/2 port is a mini-DIN plug containing six pins and is still sometimes found on all IBM compatible computers. The PS/2 mouse connector generally replaced the older DE-9 RS-232 "serial mouse" connector, while the PS/2 keyboard connector replaced the larger 5-pin/180° DIN connector used in the IBM PC/AT design.



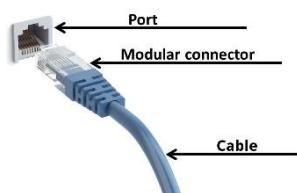
USB PORT

A USB port is a standard cable connection interface for personal computers and consumer electronics devices. USB stands for Universal Serial Bus, an industry standard for short-distance digital data communications. USB ports allow USB devices to be connected to each other with and transfer digital data over USB cables. They can also supply electric power across the cable to devices that need it. Both wired and wireless versions of the USB standard exist, although only the wired version involves USB ports and cables.



ETHERNET PORT

An Ethernet port (also called a jack or socket) is an opening on computer network equipment that Ethernet cables plug into. Their purpose is to connect wired network hardware in an Ethernet LAN, metropolitan area network (MAN), or wide area network (WAN). An Ethernet port accepts a cable that has an RJ-45 connector. The alternative to using such a cable with an Ethernet port is Wi-Fi, which eliminates the need for both the cable and the port.



Specifications:

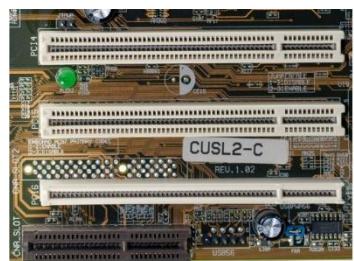
PS/2 keyboard: 1

PS/2 mouse: 1

Ethernet port:	1
USB 2.0:	4
Audio jack:	3
VGA port:	1

BUS SLOTS

A bus is a common pathway through which information flows from one computer component to another. This pathway is used for communication purposes and it is established between two or more computer components. Alternatively known as an expansion port, an expansion slot is a connection or port inside a computer on the motherboard or riser card. It provides an installation point for a hardware expansion card to be connected. For example, if you wanted to install a new video card in the computer, you'd purchase a video expansion card and install that card into the compatible expansion slot.



EXPERIMENT- 2

SPECIFICATION OF DESKTOP AND SERVER COMPUTERS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time.

8 things you should keep in mind before buying yourself a system.

1.BUDGET

Everyone has the purchasing power but how much can you really spend on buying a laptop is what you should ask yourself before purchasing one.

2.PROCESSOR

The CPU is the main part of a laptop which defines the speed of processing any task. In short, if you want to multi-task on your laptop you should buy one which has a powerful processor.

3.HARD DISK DRIVE

The hard disk drive basically keeps all your data including videos, documents, photos and other system files.

4.SCREEN SIZE

The screen size and type totally depend on what you use your laptop for.

5.DESIGN

Everyone wants to buy a laptop that is lightweight, slim, sleek and good to look at.

6.GRAPHIC PROCESSOR

7.BATTERY

A good battery life is important if you work or play on your laptop almost all the time.

8.BRAND

The brand makes a huge difference.

EXPERIMENT NO- 3

INSTALLATION OF LINUX ON VIRTUAL BOX IN WINDOWS 10

VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use. VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on.

- This is a popular method to install a Linux operating system.
- The virtual installation offers you the freedom of running Linux on an existing OS already installed on your computer.
- This means if you have Windows running, then you can just run Linux with a click of a button.
- Virtual machine software like Oracle VM can install Linux on Windows in easy steps.

STEP 1: Download and install [virtual box](#) from the official website

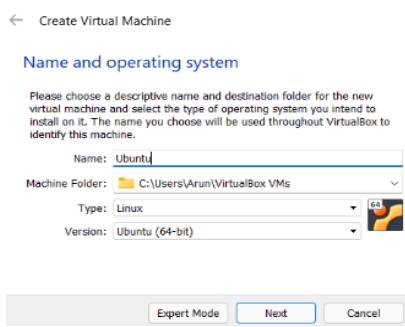


STEP 2: Download [ISO](#) file of LINUX from the official website

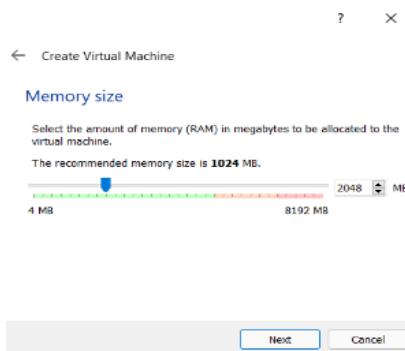


The current version of Ubuntu only works on 64-bit machines. Check the Windows Control Panel to tell if you're running a 32-bit or 64-bit system.

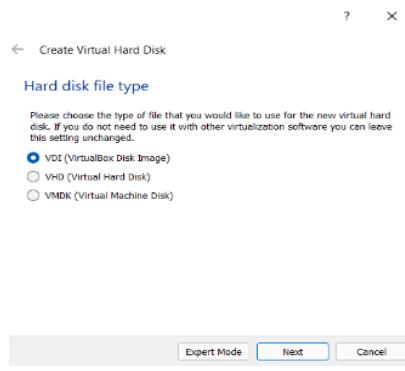
STEP 3: Open virtual box and click the new button. Give a name in the opened dialogue box and click next



STEP 4: Choose how much RAM you want to assign to the virtual machine and select Next. The recommended minimum is 1024 MB.

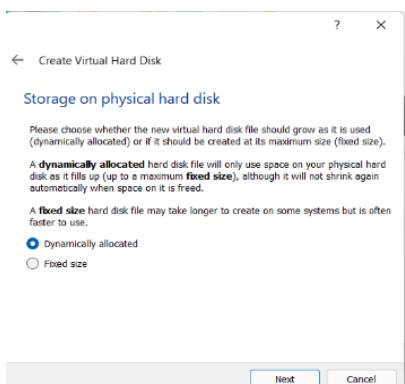


STEP 5: In the next window give VDI as the hard disk file type and click next

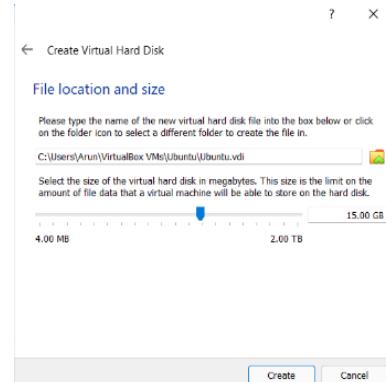


If you just want to run Ubuntu as a live image, choose **Do not add a virtual hard disk**. You must create a virtual hard drive to save the changes you make in Ubuntu

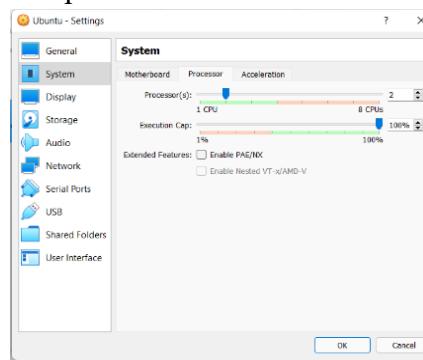
STEP 6: Give Dynamically allocated as storage on physical hard disk and click next



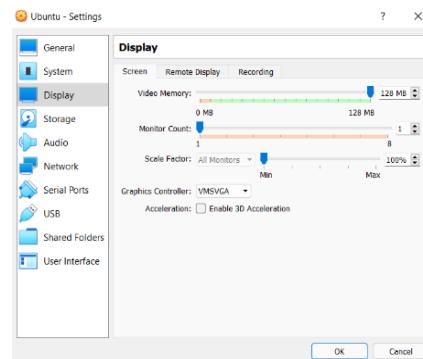
STEP 7: Provide the file location and size in the next window and click Create



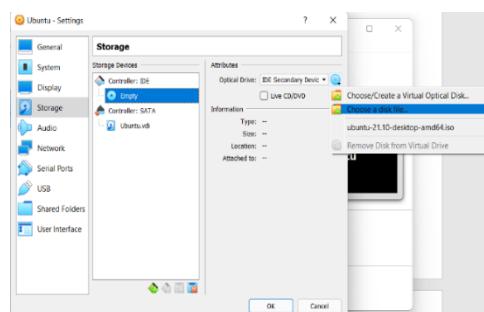
STEP 8: Now open settings, select System and then click Processor. If your system has more CPU's then change the number of processors.



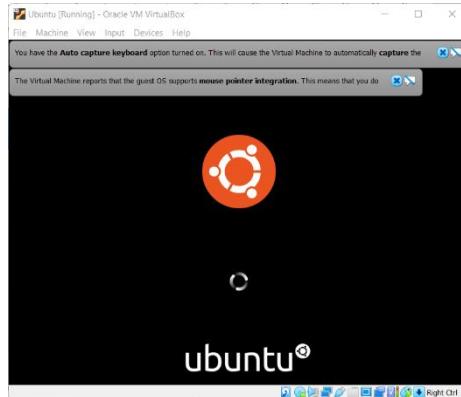
STEP 9: Select display. Now adjust the video memory for smooth performance



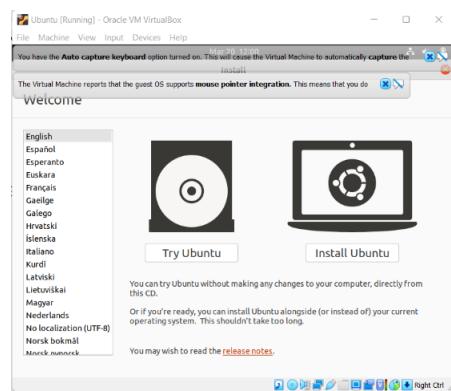
STEP 10: Now click Storage and click Empty in Storage Devices-> Controller: IDE. Then open the LINUX ISO file by clicking CD icon next to the Optical Drive dropdown box.



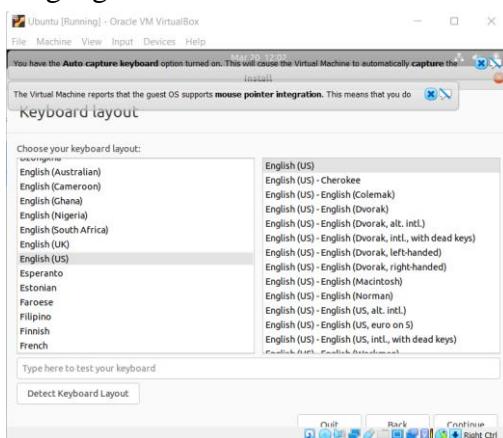
STEP 11: Now click ok and click the start button to run LINUX in virtual box.



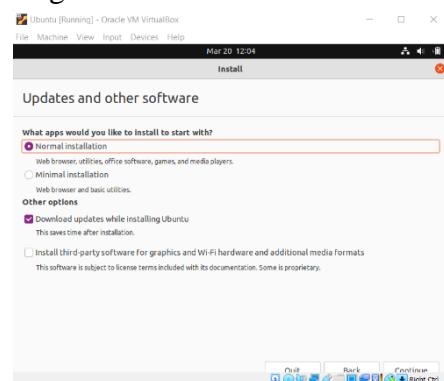
STEP 12: Click Install Ubuntu



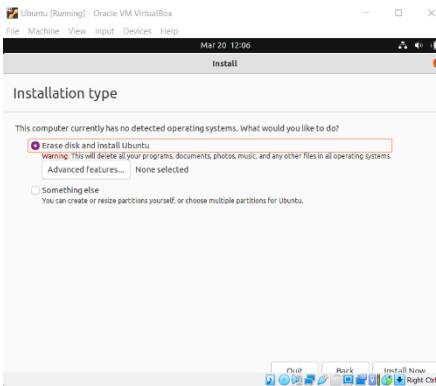
STEP 13: Select preferred language and click continue



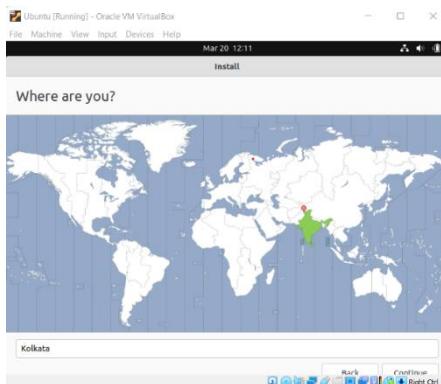
STEP 14: Select preferred settings and click continue.



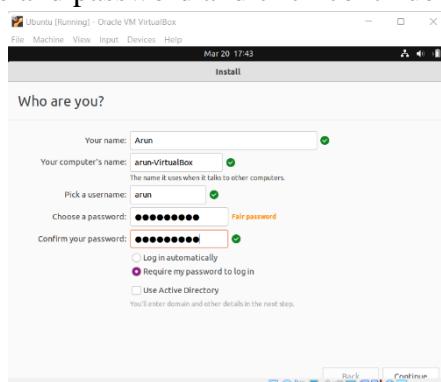
STEP 15: Select the installation type and click Install Now.



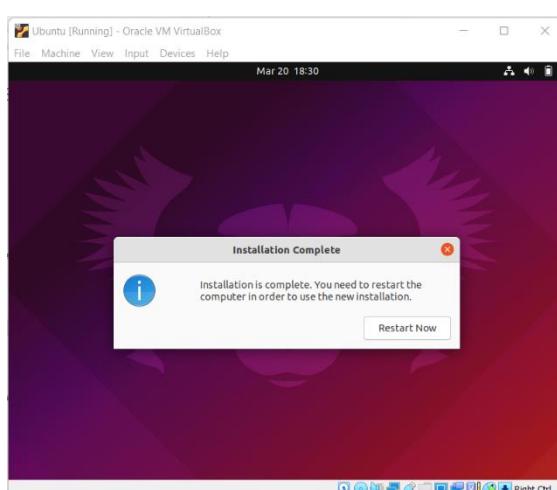
STEP 16: Select the region and click continue.



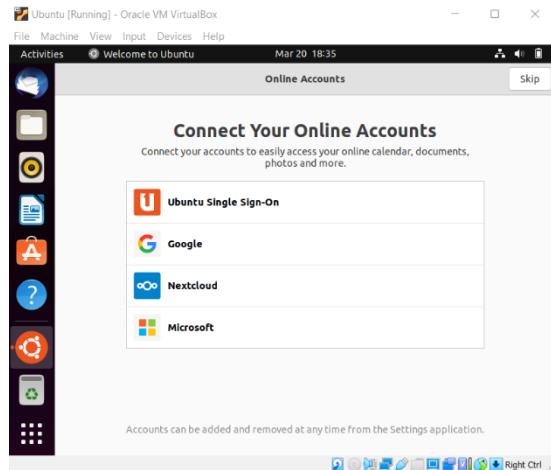
STEP 17: Now provide name and password and click continue



STEP 18: Now wait for the installation and then click Restart Now button in the dialogue box



STEP 19: Click Enter when prompted and login using the password created during the installation.



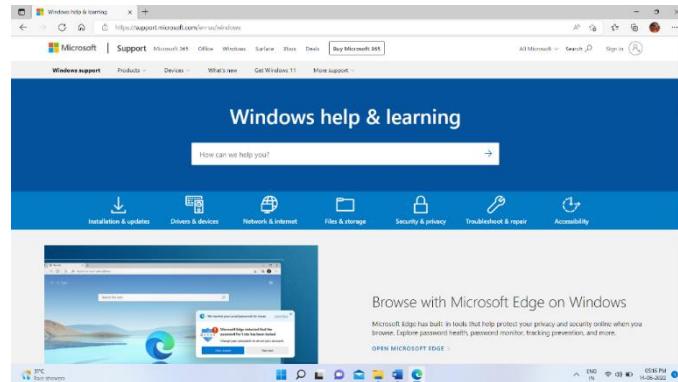
EXPERIMENT-4

INSTALLATION OF WINDOWS ON VIRTUAL BOX IN UBUNTU

STEP 1: Download and install [virtual box](#) from the official website

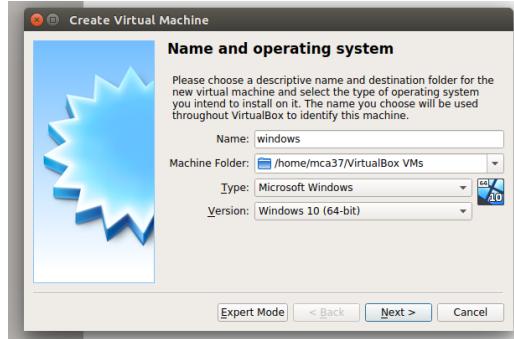


STEP 2: Download [ISO](#) file of WINDOWS from the official website When you go there, click "Get started," and follow the instructions and prompts until you finally get to the download page.

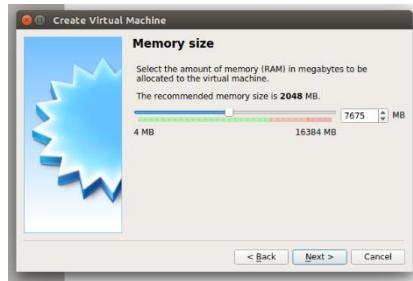


Choose your language and whether you want to download the 32-bit or 64-bit version. I downloaded the 64-bit version, because I installed it on a 64-bit machine.

STEP 3: Open virtual box and click the new button. Give a name in the opened dialogue box and click next



STEP 4: Choose how much RAM you want to assign to the virtual machine and select Next. The recommended minimum is 4 MB.



STEP 5: In the next window give VDI as the hard disk file type and click next

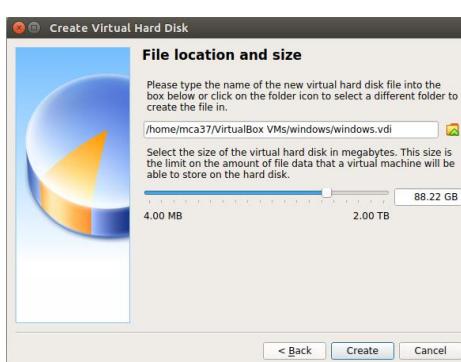


If you just want to run WINDOWS as a live image, choose **Do not add a virtual hard disk**. You must create a virtual hard drive to save the changes you make in WINDOWS.

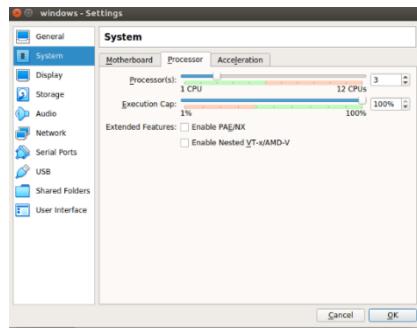
STEP 6: Give Dynamically allocated as storage on physical hard disk and click next



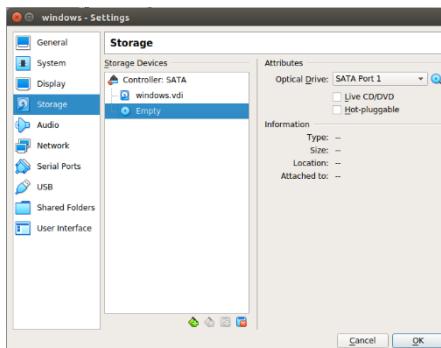
STEP 7: Provide the file location and size in the next window and click Create



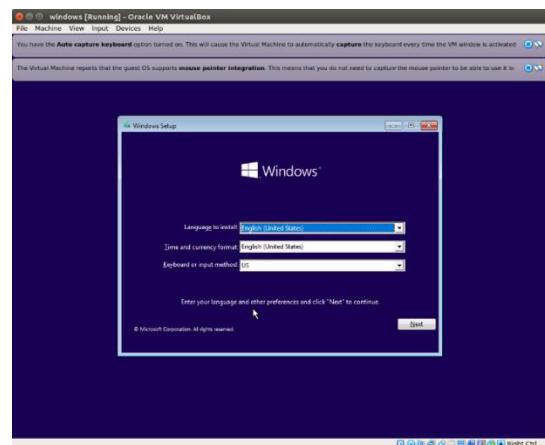
STEP 8: Now open settings, select System and then click Processor. If your system has more CPU's then change the number of processors.



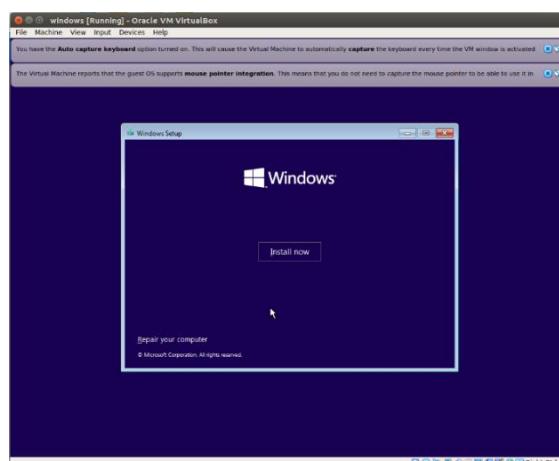
STEP 9: Now click Storage and click Empty in Storage Devices-> Controller: IDE. Then open the WINDOWS ISO file by clicking CD icon next to the Optical Drive dropdown box.



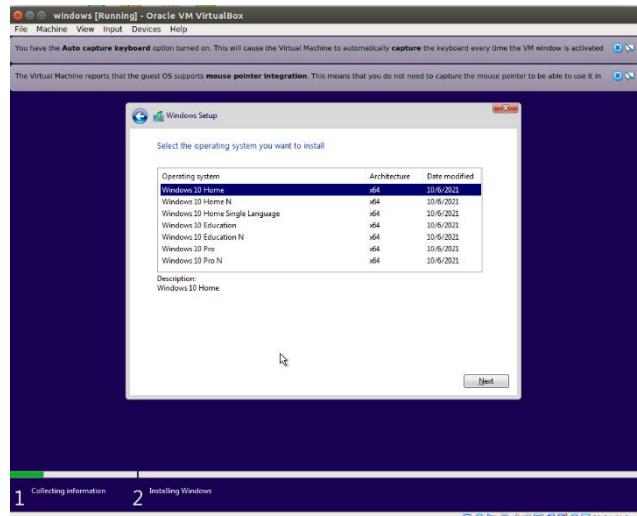
STEP 10: Now click ok and click the start button to run WINDOWS in virtual box. Select language, time format and keyboard input method and click next.



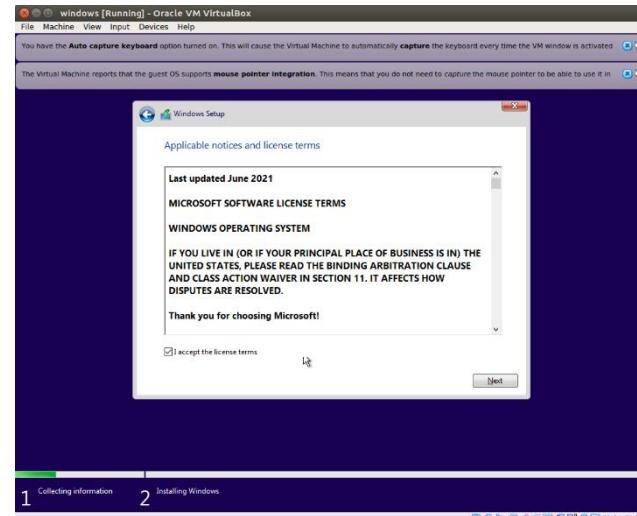
STEP 11: Now click install now.



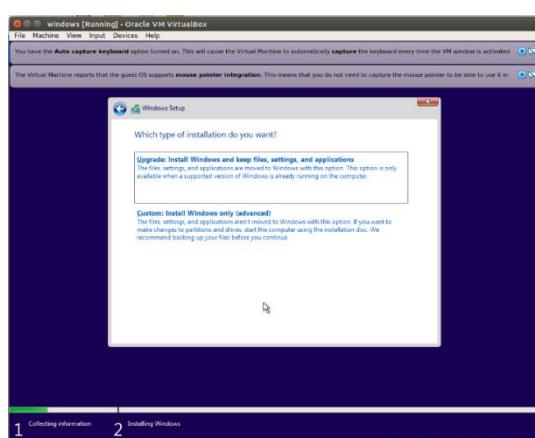
STEP 12: Select the preferred WINDOWS version from the dialog box.



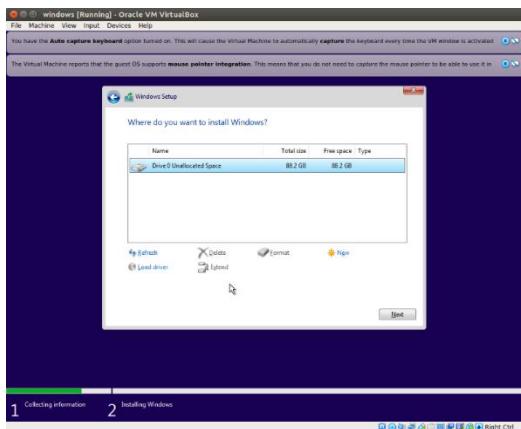
STEP 13: Accept the terms and conditions by ticking the radio button and click next.



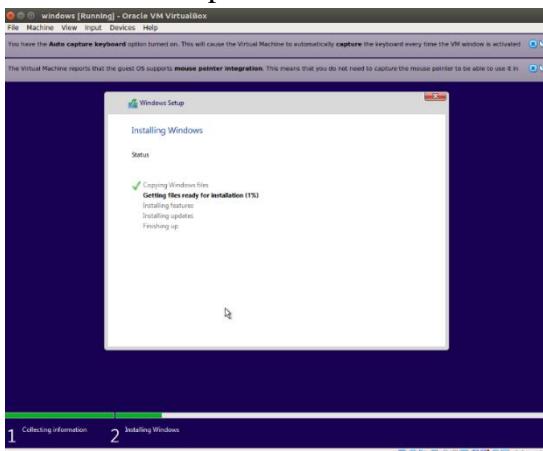
STEP 14: Choose the type of installation



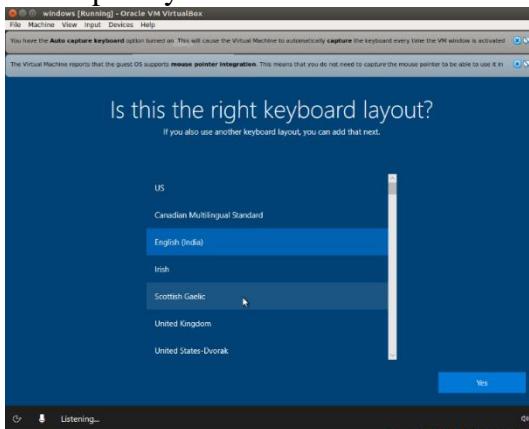
STEP 15: Choose the installation disk partition



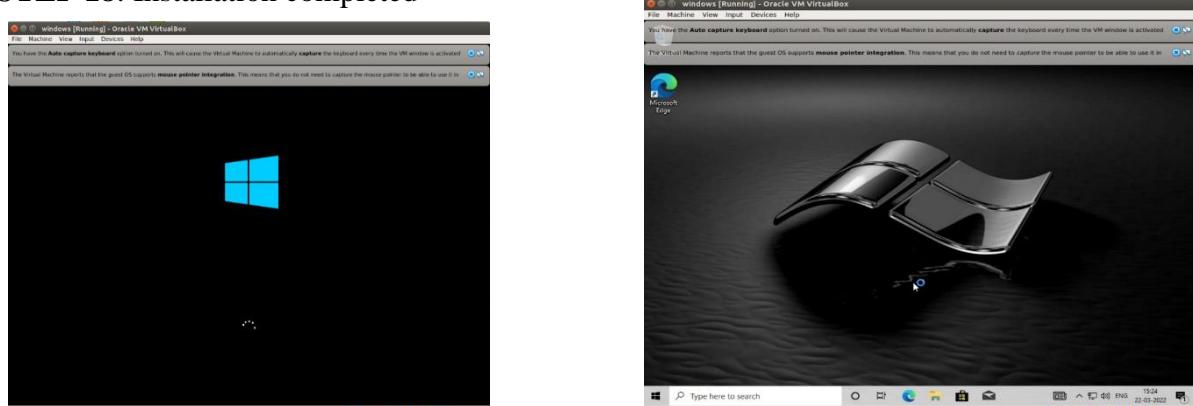
STEP 16: Wait for the installation to complete



STEP 17: Select the keyboard input layout.



STEP 18: Installation completed



EXPERIMENT-5

INSTALLATION OF WINDOWS OPERATING SYSTEM ON A BARE MACHINE

INTRODUCTION

Our Objective is to install Windows 10 Operating system to our PC/Machine via a bootable medium. We will be installing the OS from scratch assuming that we want to update/format/fix the OS in our system by making a fresh installation.

Requirements:

1. PC/Machine with at least 4GB RAM
2. An 8GB DVD or USB Flash drive.
3. Rufus software (https://rufus.ie/en_US/).
4. Required Windows 10 OS/OEM.
5. Strong stable internet connection.

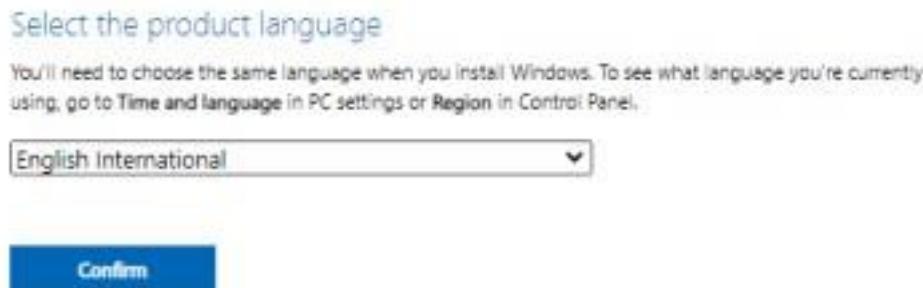
STEP 1: ACQUIRING REQUIRED OS/OEM

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/software-download/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.

Downloads

Choose a link below to begin the download. If not sure which one to choose then refer to [FAQ](#).

Windows 10 English International

[32-bit Download](#)

[64-bit Download](#)

- 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.

(https://rufus.ie/en_US/)&(<https://github.com/pbatard/rufus/releases/download/v3.14/rufus-3.14.exe>)

- 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 Rufus UI. 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 Custom Install 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.
- 9.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.
- 10.On complete installation, the PC will restart and move to an 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.
- 11.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 may have to install the drivers manually.
- 12.Monitor your system performance on Task manager, if everything is working normally, your installation is complete and your system is ready for use.

EXPERIMENT-6

INSTALLATION OF LINUX OVER WINDOWS OPERATING SYSTEM

Requirements

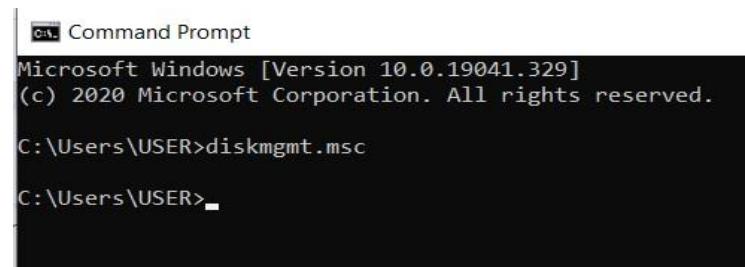
- PC with at least 2GB RAM.
- 8GB Flash drive
- Rufus
- Linux ISO file

Steps involved

- Partition a Hard Drive in Windows 10
- Make a Linux Bootable USB
- Install linux from USB

Step1: Partition a Hard Drive in Windows 10

- open cmd
- Type the command “diskmgmt.msc” and hit enter.



Command Prompt
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\USER>diskmgmt.msc
C:\Users\USER>

- Right-click on your main hard drive and select Shrink Volume.
- choose how much you want to shrink your drive.

Step2: Make a Linux Bootable USB

- Download a Linux distro in ISO format. <https://ubuntu.com/>
- Insert the USB drive into your computer.
- Download Rufus

https://rufus.ie/en_US/

- Open Rufus and select your USB drive from the Device list.
- Under Boot Selection, click the Select button and choose the ISO file you downloaded.

Step 3: Install linux from USB

- 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).
- Choose the language you wish to perform the installation and click on the Continue

- choose the first option “Normal Installation” and hit on the Continue button.
- check the Something else option and hit on the Continue button
- we’ll create our custom partition layout for Ubuntu
- hit the Install Now button in order to apply changes to disk and start the installation process.
- set your machine physical location by selecting a city nearby from the map. When done hit Continue to move ahead.
- 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.
- After the installation process reaches its end hit on the Restart Now button in order to complete the installation.
- Ubuntu has successfully installed on your system.

EXPERIMENT- 7

INSTALLATION OF SERVER SYSTEM

STEP 1: We can download the evaluation ISO of Windows Server 2012 R2 from the following link:

<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2012-r2>

STEP 2: After downloading the ISO of Microsoft, create a boot USB driver which can be created with the Microsoft tool called Windows USB/DVD Download Tool and can be downloaded from the following link:

<https://www.microsoft.com/en-us/download/windows-usb-dvd-download-tool>

STEP 3: After completing the above given steps, plug-in the USB to the server and wait for a while till it loads the files.

STEP 4: After the files are loaded, you will see the screen of language settings of installation, keyboard, time and currency format. Generally, all the default ones are also good enough to start with.

Click Next.

STEP 5: Click “Install now”.

STEP 6: Once you have clicked on Install Now, the setup will start and it will load all the files

STEP 7: Wait until the files are loaded . Let's select Windows Server 2012 DataCenter Evaluation (Server with GUI) and click Next.

STEP 8: Click “I accept the license terms” and then click on the Next button

STEP 9: A screen will appear in the window. In “Driver Options” you can create a new partition, delete or format the Hard disk. After you have done this process then you can select the partition where the Windows server 2012 R2 will be installed in our case we have one partition. Once all this is done, then click on Next.

STEP 10: Let's wait until this process finishes during this time and then the server will reboot

STEP 11: Once the reboot is done a screen will appear in the window. Set the password for the server and then click on “Finish”.

STEP 12: It will take some minutes until the setup finishes completely.

STEP 13: Once all this is done, you have completed the installation process

HOW TO SELECT THE APPROPRIATE OPERATING SYSTEM FOR YOUR SYSTEM

Stability and Robustness

Probably the most important features in an OS are stability and robustness. You are in an Internet business. You do not keep normal 9am to 5pm working hours like many conventional businesses you know. You are open 24 hours a day. You cannot afford to be off-line, for your customers will go shop at another service like yours (unless you have a monopoly :). If the OS of your choice crashes every day, first do a little investigation. There might be a simple reason which you can find and fix. There are OSs which won't work unless you reboot them twice a day. You don't want to use the OS of this kind, no matter how good the OS' vendor sales department. Do not follow flushy advertisements, follow developers' advice instead.

Memory Management

You want an OS with a good memory management, some OSs are well known as memory hogs. The same code can use twice as much memory on one OS compared to another. If the size of the mod_perl process is 10Mb and you have tens of these running, it definitely adds up!

Memory Leaks

Some OSs and/or their libraries (e.g. C runtime libraries) suffer from memory leaks. A leak is when some process requests a chunk of memory for temporary storage, but then does not subsequently release it. The chunk of memory is not then available for any purpose until the process which requested it dies. We cannot afford such leaks. A single mod_perl process sometimes serves thousands of requests before it terminates. So if a leak occurs on every request, the memory demands could become huge. Of course our code can be the cause of the memory leaks as well (check out the Apache::Leak module on CPAN). Certainly, we can reduce the number of requests to be served over the process' life, but that can degrade performance.

Sharing Memory

We want an OS with good memory sharing capabilities. As we have seen, if we preload the modules and scripts at server start-up, they are shared between the spawned children (at least for a part of a process' life - memory pages can become "dirty" and cease to be shared). This feature can reduce memory consumption a lot!

Cost and Support

If we are in a big business, we probably do not mind paying another \$1000 for some fancy OS with bundled support. But if our resources are low, we will look for cheaper and free OSs. Free does not mean bad, it can be quite the opposite. Free OSs can have the best support we can find. Some do. It is very easy to understand - most of the people are not rich and will try to use a cheaper or free OS first if it does the work for them. Since it really fits their needs, many people keep using it and eventually know it well enough to be able to provide support for others in trouble. Why would they do this for free? One reason is for the spirit of the first days of the Internet, when there was no commercial Internet and people helped each other, because someone helped them in the first place. I was there, I was touched by that spirit and I am keen to keep that spirit alive.

Discontinued Products

The OSs in this hazard group tend to be developed by a single company or organization.

You might find yourself in a position where you have invested a lot of time and money into developing some proprietary software that is bundled with the OS you chose (say writing a mod_perl handler which takes advantage of some proprietary features of the OS and which will not run on any other OS). Things are under control; the performance is great and you sing with happiness on your way to work. Then, one day, the company which supplies your beloved OS goes bankrupt (not unlikely nowadays), or they produce a newer incompatible version and they will not support the old one (happens all the time). You are stuck with their early masterpiece, no support and no source code! What are you going to do? Invest more money into porting the software to another OS.

OS Releases

Actively developed OSs generally try to keep pace with the latest technology developments, and continually optimize the kernel and other parts of the OS to become better and faster. Nowadays, the Internet and networking in general are the hottest topics for system developers. Sometimes a simple OS upgrade to the latest stable version can save you an expensive hardware upgrade. Also, remember that when you buy new hardware, chances are that the latest software will make the most of it.

CYCLE-2

EXPERIMENT- 8

STUDY ON COMMAND LINE TEXT EDITORS

LINUX TEXT EDITORS

Text editors are programs used to create and edit plain text files. The main purpose of the text editor is to create a file to be used by another program. A text editor plays an important role while coding. So, it is important to select the best text editor. A text editor should not only be simple but also functional and should be good to work with.

A text editor with IDE features is considered as a good text editor.

Vi/VIM EDITOR

Vim editor is one of the most used and powerful command-line based editor of the Linux system. By default, it is supported by most Linux distros. It has enhanced functionalities of the old Unix Vi editor. It is a user-friendly editor and provides the same environment for all the Linux distros. It is also termed as programmer's editor because most programmers prefer Vi editor.

Vi editor has some special features such as Vi modes and syntax highlighting that makes it powerful than other text editors. Generally, it has two modes:

Command Mode: The command mode allows us to perform actions on files. By default, it starts in command mode. In this mode, all types of words are considered as commands. We can execute commands in this mode.

Insert Mode: The insert mode allows to insert text on files. To switch from command mode to insert mode, press the Esc key to exit from active mode and 'i' key.

NANO EDITOR

Nano is a straightforward editor. It is designed for both beginners and advanced users. In the nano editor, the useful options are given at the bottom, use the CTRL+ option to perform an operation.

Gedit Editor

Gedit editor is the default editor for the GNOME desktop environment. When we open a file, it will open with the Gedit editor. It provides straightforward functionalities like any basic text editor. It is a lightweight editor with a straightforward user interface.

Sublime Text

The sublime text editor is also one of the most popular IDE-based text editors. It is used as a development environment tool more than a text editor.

VSCode Editor

VSCode editor is a modern and widely used text editor. It is built by Microsoft and has support for Linux, Mac and Windows OS. It is facilitated with many powerful features to support many programming languages and markup languages. Pico Editor

The Pico editor is a terminal-based Linux text editor. It has built-in support for pine news and email clients.

GNU Emacs

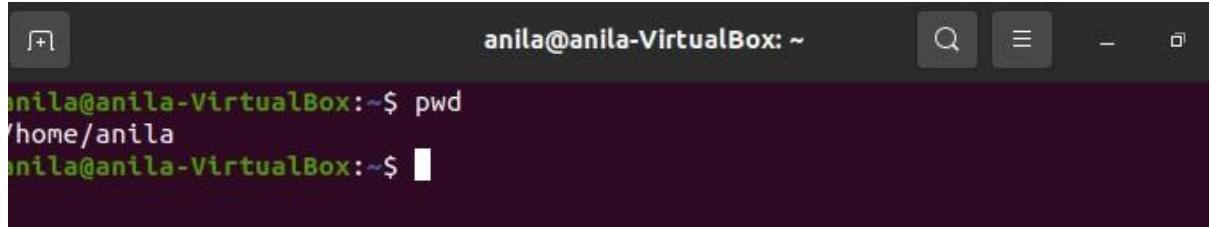
GNU Emacs is the oldest and simplest text editor for the Linux system. It is a part of the GNU project. It is still a popular text editor used by thousands of users because of its simplicity. It is written in C and LISP programming languages.

EXPERIMENT- 9

BASIC LINUX COMMANDS

1. pwd

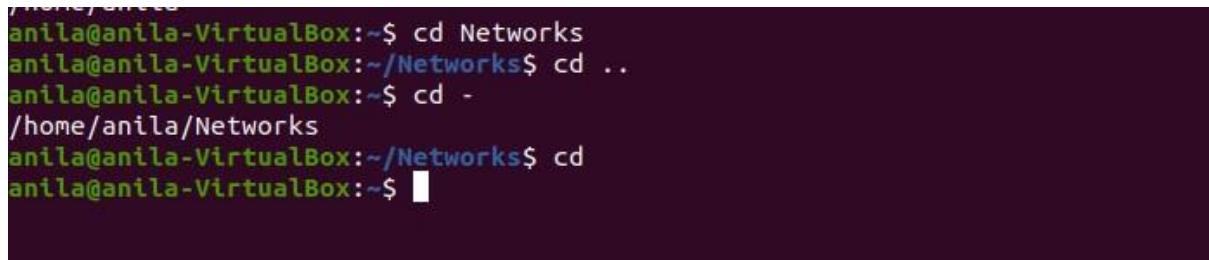
Use the pwd command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/).



```
anila@anila-VirtualBox:~$ pwd
/home/anila
anila@anila-VirtualBox:~$
```

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.



```
/home/anila$ cd Networks
anila@anila-VirtualBox:~/Networks$ cd ..
anila@anila-VirtualBox:~/Networks$ cd -
/home/anila/Networks
anila@anila-VirtualBox:~/Networks$ cd
anila@anila-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.

- 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.

```

anila@anila-VirtualBox:~$ ls
acn.txt  Documents  Music      Pictures  Templates
Desktop  Downloads  Networks  Public    Videos
anila@anila-VirtualBox:~$ ls -R
.:
acn.txt  Documents  Music      Pictures  Templates
Desktop  Downloads  Networks  Public    Videos

./Desktop:
./Documents:
./Downloads:
./Music:
./Networks:
./Pictures:
./Public:
./Templates:

```



```

anila@anila-VirtualBox:~$ ls -a
..          .bashrc  Downloads  Music      .ssh
.acn.txt    .cache   .emacs.d  Networks  .sudo_as_admin_successful
.bash_history .config   .gnupg   Pictures  Templates
.bash_logout  Desktop  .local   .profile  Videos
.anila@anila-VirtualBox:~$ ls -al
total 92
drwxr-xr-x 18 anila anila 4096 Sep 19 22:30 .
drwxr-xr-x  3 root  root  4096 Sep 18 00:38 ..
-rw-rw-r--  1 anila anila 107 Sep 19 20:31 acn.txt
-rw-r----- 1 anila anila  82 Sep 19 22:29 .bash_history
-rw-r--r--  1 anila anila 220 Sep 18 00:38 .bash_logout
-rw-r--r--  1 anila anila 3771 Sep 18 00:38 .bashrc
drwx----- 13 anila anila 4096 Sep 18 20:22 .cache
drwx----- 12 anila anila 4096 Sep 19 20:34 .config
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Desktop
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Documents
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Downloads
drwx----- 3 anila anila 4096 Sep 19 20:27 .emacs.d
drwx----- 3 anila anila 4096 Sep 19 19:58 .gnupg
drwxr-xr-x  3 anila anila 4096 Sep 18 01:13 .local
drwx----- 5 anila anila 4096 Sep 18 20:22 .mozilla
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Music
drwxrwxr-x  2 anila anila 4096 Sep 19 22:29 Networks
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Pictures
-rw-r--r--  1 anila anila  807 Sep 18 00:38 .profile
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Public
drwx----- 2 anila anila 4096 Sep 18 01:22 ssh

```



```

anila@anila-VirtualBox:~$ ls -t
-rw-r--r--  1 anila anila 3771 Sep 18 00:38 .bashrc
drwx----- 13 anila anila 4096 Sep 18 20:22 .cache
drwx----- 12 anila anila 4096 Sep 19 20:34 .config
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Desktop
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Documents
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Downloads
drwx----- 3 anila anila 4096 Sep 19 20:27 .emacs.d
drwx----- 3 anila anila 4096 Sep 19 19:58 .gnupg
drwxr-xr-x  3 anila anila 4096 Sep 18 01:13 .local
drwx----- 5 anila anila 4096 Sep 18 20:22 .mozilla
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Music
drwxrwxr-x  2 anila anila 4096 Sep 19 22:29 Networks
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Pictures
-rw-r--r--  1 anila anila  807 Sep 18 00:38 .profile
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Public
drwx----- 2 anila anila 4096 Sep 18 01:22 ssh
-rw-r--r--  1 anila anila  0 Sep 19 20:09 .sudo_as_admin_successful
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Templates
drwxr-xr-x  2 anila anila 4096 Sep 18 01:13 Videos
anila@anila-VirtualBox:~$ ls -R
Networks  Desktop  Downloads  Pictures  Templates
acn.txt  Documents  Music      Public    Videos
anila@anila-VirtualBox:~$ ls -r
Videos  Public  Networks  Downloads  Desktop
Templates  Pictures  Music      Documents  acn.txt
anila@anila-VirtualBox:~$ ls -tr
Videos  Public  Music      Documents  acn.txt
Templates  Pictures  Downloads  Desktop  Networks
anila@anila-VirtualBox:~$ 

```

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.

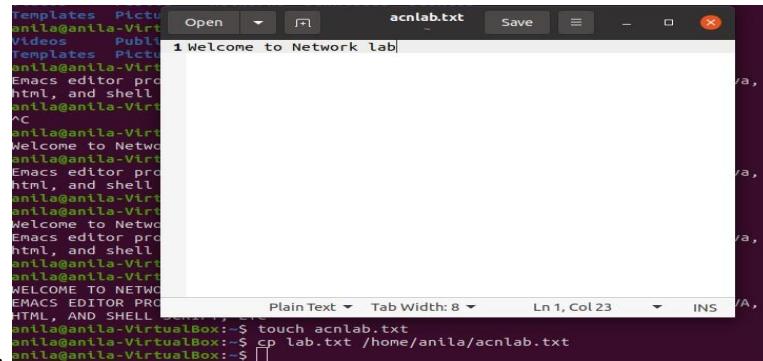
- cat > filename creates a new file
- cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, cat filename | tr a-z A-Z >output.txt

```
anila@anila-VirtualBox:~$ cat acn.txt
Emacs editor provide environment to programmers develop program such as c,java,
html, and shell script, etc
anila@anila-VirtualBox:~$ cat >lab.txt
^C
anila@anila-VirtualBox:~$ cat lab.txt
Welcome to Network lab
anila@anila-VirtualBox:~$ cat acn.txt
Emacs editor provide environment to programmers develop program such as c,java,
html, and shell script, etc
anila@anila-VirtualBox:~$ cat lab.txt acn.txt>new.txt
anila@anila-VirtualBox:~$ cat new.txt
Welcome to Network lab
Emacs editor provide environment to programmers develop program such as c,java,
html, and shell script, etc
anila@anila-VirtualBox:~$ cat new.txt | tr a-z A-Z>output.txt
anila@anila-VirtualBox:~$ cat output.txt
WELCOME TO NETWORK LAB
EMACS EDITOR PROVIDE ENVIRONMENT TO PROGRAMMERS DEVELOP PROGRAM SUCH AS C, JAVA,
HTML, AND SHELL SCRIPT, ETC
anila@anila-VirtualBox:~$
```

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



```
Templates Pictures Open Save acnlab.txt - - - - - 
Videos Public acnlab.txt - - - - - 
anila@anila-Virt Welcome to Network lab 
anila@anila-Virt Emacs editor pro 
html, and shell 
anila@anila-Virt 
anila@anila-Virt Welcome to Network lab 
Emacs editor pro 
html, and shell 
anila@anila-Virt 
anila@anila-Virt WELCOME TO NETWORK LAB 
EMACS EDITOR PROVIDED ENVIRONMENT TO PROGRAMMERS DEVELOP PROGRAM SUCH AS C, JAVA, 
HTML, AND SHELL SCRIPT, ETC 
anila@anila-VirtualBox:~$ touch acnlab.txt 
anila@anila-VirtualBox:~$ cp lab.txt /home/anila/acnlab.txt 
anila@anila-VirtualBox:~$
```

```
html, and shell script, etc
anila@anila-VirtualBox:~$ cat lab.txt acn.txt>new.txt
anila@anila-VirtualBox:~$ cat new.txt
Welcome to Network lab
Emacs editor provide environment to programmers develop program such as c,java,
html, and shell script, etc
anila@anila-VirtualBox:~$ cat new.txt | tr a-z A-Z>output.txt
anila@anila-VirtualBox:~$ cat output.txt
WELCOME TO NETWORK LAB
EMACS EDITOR PROVIDE ENVIRONMENT TO PROGRAMMERS DEVELOP PROGRAM SUCH AS C, JAVA,
HTML, AND SHELL SCRIPT, ETC
anila@anila-VirtualBox:~$ touch acnlab.txt
anila@anila-VirtualBox:~$ cp lab.txt /home/anila/acnlab.txt
anila@anila-VirtualBox:~$ cp -i lab.txt acnlab.txt
cp: overwrite 'acnlab.txt'? y
anila@anila-VirtualBox:~$ cp -r Networks Acnnew
anila@anila-VirtualBox:~$ cd Acnnew
anila@anila-VirtualBox:~/Acnnew$ ls
Networks
anila@anila-VirtualBox:~/Acnnew$ cd ..
anila@anila-VirtualBox:~$ cd Networks
anila@anila-VirtualBox:~/Networks$ touch lab2.txt
anila@anila-VirtualBox:~/Networks$ cp -u lab.txt lab2.txt
anila@anila-VirtualBox:~/Networks$ cat lab2.txt
cat: lab2.txt: No such file or directory
anila@anila-VirtualBox:~/Networks$ cat lab2.txt
anila@anila-VirtualBox:~/Networks$ cp -u lab.txt lab2.txt
anila@anila-VirtualBox:~/Networks$ cat lab2.txt
```

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.

To rename files, the Linux is mv oldname.text newname.text

```
anila@anila-VirtualBox:~$ ls
acnlab.txt acn.txt Documents Music      new.txt    Pictures  Templates
Acnew      Desktop Downloads Networks  output.txt  Public    Videos
anila@anila-VirtualBox:~$ mv output.txt Acnew
anila@anila-VirtualBox:~$ ls
acnlab.txt acn.txt Documents Music      new.txt    Public    Videos
Acnew      Desktop Downloads Networks  Pictures   Templates
anila@anila-VirtualBox:~$ mv Acnew labnew
anila@anila-VirtualBox:~$ ls
acnlab.txt Desktop  Downloads Music      new.txt    Public    Videos
acn.txt   Documents labnew   Networks  Pictures   Templates
anila@anila-VirtualBox:~$
```

7. mkdir

Use mkdir command to make a new directory

```
anila@anila-VirtualBox:~$ mkdir New1
anila@anila-VirtualBox:~$ ls
acnlab.txt Desktop  Downloads Music      New1     Pictures  Templates
acn.txt   Documents labnew   Networks  new.txt  Public    Videos
anila@anila-VirtualBox:~$
```

8. rmdir

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

```
anila@anila-VirtualBox:~$ rmdir New1
anila@anila-VirtualBox:~$ ls
acnlab.txt Desktop  Downloads Music      new.txt  Public    Videos
acn.txt   Documents labnew   Networks  Pictures  Templates
anila@anila-VirtualBox:~$
```

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.

```
anila@anila-VirtualBox:~$ rm -r labnew
anila@anila-VirtualBox:~$ ls
acnlab.txt Desktop  Downloads Networks Pictures  Templates
acn.txt   Documents Music      new.txt  Public    Videos
anila@anila-VirtualBox:~$
```

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 an HTML file entitled Web under the Documents directory.

```
anila@anila-VirtualBox:~$ touch hai.txt
anila@anila-VirtualBox:~$ ls
acnlab.txt Desktop  Downloads Music      new.txt  Public    Videos
acn.txt   Documents hai.txt   Networks  Pictures  Templates
anila@anila-VirtualBox:~$
```

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.

```
Processing triggers for man-db (2.9.1-1) ...
anila@anila-VirtualBox:~$ locate Newfile
/home/anila/Networks/Newfile
```

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.

```
anila@anila-VirtualBox:~$ find -type d -name Newfile
./Networks/Newfile
anila@anila-VirtualBox:~$ find -name Networks
./Networks
```

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.

```
grep: lab: No such file or directory
anila@anila-VirtualBox:~/Networks$ grep Welcome lab.txt
Welcome to Network lab
anila@anila-VirtualBox:~/Networks$
```

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.

```
anila@anila-VirtualBox:~/Networks$ sudo apt install git
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
git-man liberror-perl
Suggested packages:
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:
git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 310 not upgraded.
Need to get 5,465 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

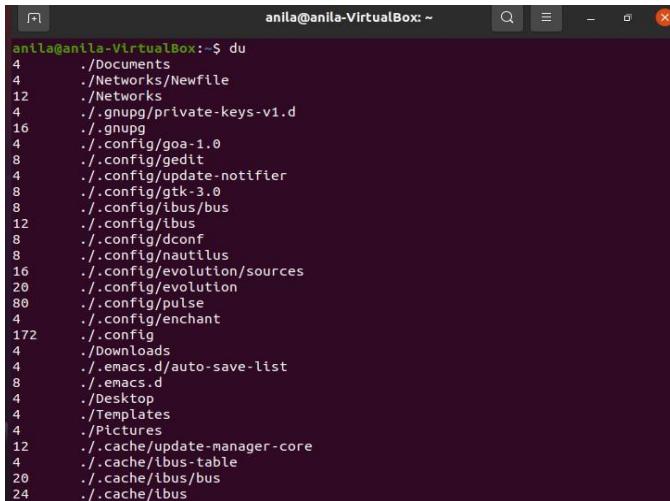
15. df

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.

```
anila@anila-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             987300      0   987300  0% /dev
tmpfs            203272   1328  201944  1% /run
/dev/sda5       35342912 9279900  24237988 28% /
tmpfs           1016356      0  1016356  0% /dev/shm
tmpfs             5120       4   5116  1% /run/lock
tmpfs           1016356      0  1016356  0% /sys/fs/cgroup
/dev/loop0        56832   56832      0 100% /snap/core18/1988
/dev/loop2        56832   56832      0 100% /snap/core18/2128
/dev/loop1       224256   224256      0 100% /snap/gnome-3-34-1804/66
/dev/loop3       224256   224256      0 100% /snap/gnome-3-34-1804/72
/dev/loop5       66688   66688      0 100% /snap/gtk-common-themes/1515
/dev/loop7       33152   33152      0 100% /snap/snapd/12883
/dev/loop6        52352   52352      0 100% /snap/snap-store/518
/dev/loop4        66432   66432      0 100% /snap/gtk-common-themes/1514
/dev/loop9        52224   52224      0 100% /snap/snap-store/547
/dev/loop8        31872   31872      0 100% /snap/snapd/11036
/dev/sda1       523248       4  523244  1% /boot/efi
tmpfs            203268     28  203240  1% /run/user/1000
anila@anila-VirtualBox:~$
```

16. du

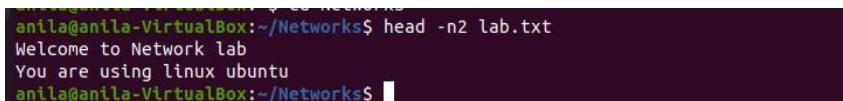
If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.



```
anila@anila-VirtualBox:~$ du
4      ./Documents
4      ./Networks/Newfile
12     ./Networks
4      ./gnupg/private-keys-v1.d
16     ./gnupg
4      ./config/goa-1.0
8      ./config/gedit
4      ./config/update-notifier
8      ./config/gtk-3.0
8      ./config/ibus/bus
12     ./config/ibus
8      ./config/dconf
8      ./config/nautilus
16     ./config/evolution/sources
20     ./config/evolution
80     ./config/pulse
4      ./config/enchant
172    ./config
4      ./Downloads
4      ./emacs.d/auto-save-list
8      ./emacs.d
4      ./Desktop
4      ./Templates
4      ./Pictures
12     ./cache/update-manager-core
4      ./cache/ibus-table
20     ./cache/ibus/bus
24     ./cache/ibus
```

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. Show the first five lines, type head -n 5



```
anila@anila-VirtualBox:~/Networks$ head -n2 lab.txt
Welcome to Network lab
You are using linux ubuntu
anila@anila-VirtualBox:~/Networks$
```

18. tail

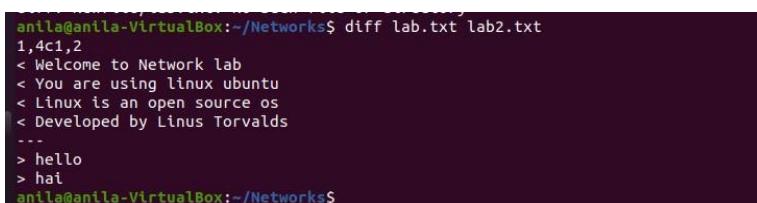
This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. For example, tail -n filenamext.



```
anila@anila-VirtualBox:~/Networks$ tail -n3 lab.txt
You are using linux ubuntu
Linux is an open source os
Developed by Linus Torvalds
anila@anila-VirtualBox:~/Networks$
```

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.



```
anila@anila-VirtualBox:~/Networks$ diff lab.txt lab2.txt
1,4c1,2
< Welcome to Network lab
< You are using linux ubuntu
< Linux is an open source os
< Developed by Linus Torvalds
---
> hello
> hai
anila@anila-VirtualBox:~/Networks$
```

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.

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 .

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.

23. ps

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

```
> ncat  
anila@anila-VirtualBox:~/Networks$ ps  
 PID TTY      TIME CMD  
 1820 pts/0    00:00:00 bash  
 4238 pts/0    00:00:00 ps  
anila@anila-VirtualBox:~/Networks$
```

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.

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.

```
anila@anila-VirtualBox:~/Networks$ ping google.com
PING google.com (142.250.193.110) 56(84) bytes of data.
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=1 ttl=116 time=190 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=2 ttl=116 time=446 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=3 ttl=116 time=49.6 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=4 ttl=116 time=111 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=5 ttl=116 time=113 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=6 ttl=116 time=218 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=7 ttl=116 time=284 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=8 ttl=116 time=49.2 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=9 ttl=116 time=68.2 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=10 ttl=116 time=82.3 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=11 ttl=116 time=182 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=12 ttl=116 time=81.1 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=13 ttl=116 time=53.6 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=14 ttl=116
```

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.

```
anila@anila-VirtualBox:~/Networks$ wget https://www.oracle.com/in/index.html
--2021-09-20 23:54:05-- https://www.oracle.com/in/index.html
Resolving www.oracle.com (www.oracle.com)... 23.9.17.11, 2600:140f:3:1186::a15,
2600:140f:3:119d::a15
Connecting to www.oracle.com (www.oracle.com)|23.9.17.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html [=> .....] 42.22K --.-KB/s in 0.06s

2021-09-20 23:54:06 (692 KB/s) - 'index.html' saved [43229]

anila@anila-VirtualBox:~/Networks$
```

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.

```
anila@anila-VirtualBox:~/Networks$ uname
Linux
```

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.

```
anila@anila-VirtualBox:~/Networks$ top
top - 00:18:29 up 1:17, 1 user, load average: 0.25, 0.09, 0.08
Tasks: 183 total, 2 running, 181 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.3 us, 0.7 sy, 0.0 ni, 96.5 id, 0.2 wa, 0.0 hi, 0.3 si, 0.0 st
Mem: 1985.1 total, 579.1 free, 640.3 used, 765.7 buff/cache
Swap: 1634.2 total, 1634.2 free, 0.0 used. 1182.9 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
1488 anila     20   0 4199088 327264 119452 R  2.6 16.1 2:13.07 gnome+-+
1283 anila     20   0 824224 59324 39248 S  2.3 2.9 0:59.42 Xorg
1812 anila     20   0 823332 51088 38612 S  1.7 2.5 0:27.25 gnome+-+
4269 anila     20   0 20484 3784 3140 R  0.3 0.2 0:16.50 top
  1 root      20   0 167600 11696 8572 S  0.0 0.6 0:05.99 systemd
  2 root      20   0      0      0  0 S  0.0 0.0 0:00.00 kthread+
  3 root      0 -20      0      0  0 I  0.0 0.0 0:00.00 rcu_gp
  4 root      0 -20      0      0  0 I  0.0 0.0 0:00.00 rcu_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 0:00.00 rcu_ta+
 11 root     20   0      0      0  0 S  0.0 0.0 0:00.00 rcu_ta+
 12 root     20   0      0      0  0 S  0.0 0.0 0:00.52 ksoftti+
 13 root     20   0      0      0  0 I  0.0 0.0 0:01.71 rcu_sc+
 14 root      rt  0      0      0  0 S  0.0 0.0 0:00.05 migrat+
 15 root     -51  0      0      0  0 S  0.0 0.0 0:00.00 idle_i+
 16 root     20   0      0      0  0 S  0.0 0.0 0:00.00 cpuhp/0
 17 root     20   0      0      0  0 S  0.0 0.0 0:00.00 cpuhp/1
 18 root     -51  0      0      0  0 S  0.0 0.0 0:00.00 idle_i+
 19 root      rt  0      0      0  0 S  0.0 0.0 0:00.70 migrat+
```

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.

```
anila@anila-VirtualBox:~/Networks$ history
1 sudo apt-get install emacs
2 emacs --version
3 emacs
4 emacs acn.txt
5 pwd
6 cd Networkslab
7 cd ..
8 cd
9 cd Networks
10 grep Welcome lab
11 grep Welcome lab.txt
12 sudo apt install git
13 cd ..
14 df
15 du
16 cd ..
17 cd
18 cd Networks
19 head -n2 lab.txt
20 tail -n3 lab.txt
21 diff lab.txt Newfile
22 diff lab.txt lab2.txt
23 ps
24 ping google.com
25 ping google.com
26 wget https://www.oracle.com/in/index.html
27 wget https://www.oracle.com/in/index.html
28 uname
```

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.

```
NETWORKS(5)          Linux System Administration          NETWORKS(5)

NAME
    networks - network name information

DESCRIPTION
    The file /etc/networks is a plain ASCII file that describes known
    DARPA networks and symbolic names for these networks. Each line repre-
    sents a network and has the following structure:

        name number aliases ...
    where the fields are delimited by spaces or tabs. Empty lines are ig-
    nored. The hash character (#) indicates the start of a comment: this
    character, and the remaining characters up to the end of the current
    line, are ignored by library functions that process the file.

    The field descriptions are:

    name   The symbolic name for the network. Network names can contain
              any printable characters except white-space characters or the
              comment character.

    number  The official number for this network in numbers-and-dots nota-
              tion (see inet(3)). The trailing ".0" (for the host component
              of the network address) may be omitted.

    aliases
    Manual page networks(5) line 1 (press h for help or q to quit)
```

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

```
anila@anila-VirtualBox:~$ echo haihello  
haihello
```

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.

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.

```
anila@anila-VirtualBox:~$ hostname  
anila-VirtualBox  
anila@anila-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.

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

EXPERIMENT- 10

FAMILIARISATION WITH VI EDITOR

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor –

- 1) It is available in almost all Linux Distributions
- 2) It works the same across different platforms and Distributions
- 3) It is user-friendly. Hence, millions of Linux use it for their editing needs

Nowadays, there are advanced versions of the vi-editor available, and the most popular one is VIM which is Vi Improved. Some of the other ones are Elvis, Nvi, Nano, and Vile.

1.VI OPERATIONAL MODES

- Vi Command Mode:
 - The vi editor opens in this mode, and it only understands commands.
 - In this mode, you can perform administrative tasks such as move the cursor and cut, copy, paste the text.
 - This mode also saves the changes you have made to the file.
 - Commands are case sensitive. The right letter case should be used.
- Vi Insert Mode:
 - This mode is for inserting text in the file.
 - You can switch to the Insert mode from the command mode by pressing 'i' on the keyboard
 - Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
 - To return to the command mode and save the changes you have made you need to press the Esc key

2. STARTING VI

To open a file with vi, type: vi 'filename'.txt0

- a. If the file does not exist, a screen will appear with just a cursor at the top followed by tildes (~) in the first column.
- b. If the file does exist, the first few lines of the file will appear.
- c. The status line at the bottom of the screen shows error messages and provides information and feedback, including the name of the file.

3. Vi EDITING COMMANDS

- ★ i - Insert at cursor (goes into insert mode)
- ★ a - Write after cursor (goes into insert mode)
- ★ A - Write at the end of line (goes into insert mode)
- ★ ESC - Terminate insert mode

- ★ u - Undo last change
- ★ U - Undo all changes to the entire line
- ★ - Open a new line (goes into insert mode)
- ★ dd - Delete line
- ★ 3dd - Delete 3 lines.
- ★ D - Delete contents of line after the cursor
- ★ C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
- ★ dw - Delete word
- ★ 4dw - Delete 4 words
- ★ cw - Change word
- ★ x - Delete character at the cursor
- ★ r - Replace character
- ★ R - Overwrite characters from cursor onward
- ★ s - Substitute one character under cursor continue to insert
- ★ S - Substitute entire line and begin to insert at the beginning of the line
- ★ ~ - Change case of individual character

Note: You should be in the "command mode" to execute these commands. VI editor is case-sensitive so make sure you type the commands in the right letter-case.

Make sure you press the right command otherwise you will end up making undesirable changes to the file. You can also enter the insert mode by pressing a, A, o, as required.

Moving within a file

- ★ k - Move cursor up
- ★ j - Move cursor down
- ★ h - Move cursor left
- ★ l - Move cursor right

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can also use the arrow keys on the keyboard.

Saving and Closing the file

- ★ Shift+zz - Save the file and quit
- ★ :w - Save the file but keep it open
- ★ :q - Quit without saving
- ★ :wq - Save the file and quit

You should be in the command mode to exit the editor and save changes to the file.

WORKING WITH Vi EDITOR

i-Use this command to insert text before the current cursor location o-Use this command to create a new line for text below the current cursor location yy-Use this command to copy the current line

P-Use this command to paste the copied text after the cursor x-Use this command to delete the character under the current location.

EXPERIMENT- 11

FAMILIARISATION WITH EMACS EDITOR

- 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 run 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.

Some Features:

- Faster
- Powerful
- Simple

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)

Some Important commands

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

Some Cursor Movement Command

- 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

Programming in emacs:

The emacs editor provides an environment to programmers develop programs such as c, java, HTML and shell script etc.

After writing the program source code, it also compiles, links, debug and executes itself in the terminal window to test it.

Execution of emacs on Linux:

Installing emacs editor on Ubuntu

```
$ sudo apt-get install emacs
```

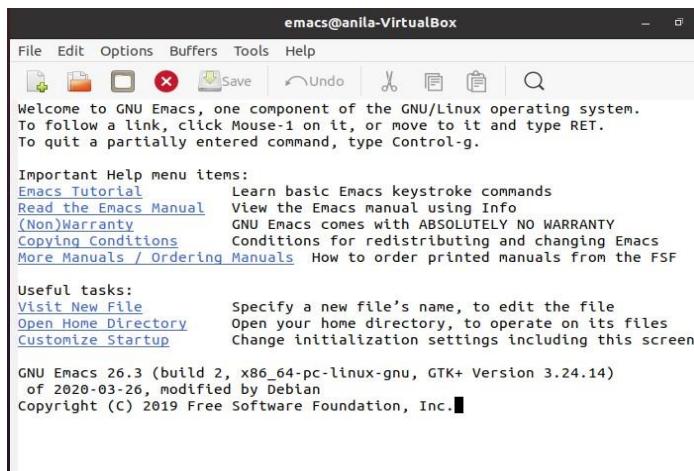
```
[sudo] password for anila:
sudo: 1 incorrect password attempt
anila@anila-VirtualBox:~$ sudo apt-get install emacs
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts
  imagemagick-6-common liblqr-1-0 libm17n-0 libmagickcore-6.q16-6
  libmagickwand-6.q16-6 libotf0 libwebpmux3 m17n-db
Suggested packages:
  mailutils emacs-common-non-dfsg ncurses-term m17n-docs
  libmagickcore-6.q16-6-extra gawk
The following NEW packages will be installed:
  emacs emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts
  imagemagick-6-common liblqr-1-0 libm17n-0 libmagickcore-6.q16-6
  libmagickwand-6.q16-6 libotf0 m17n-db
The following packages will be upgraded:
  libwebpmux3
1 upgraded, 13 newly installed, 0 to remove and 310 not upgraded.
Need to get 39.6 MB of archives.
After this operation, 151 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

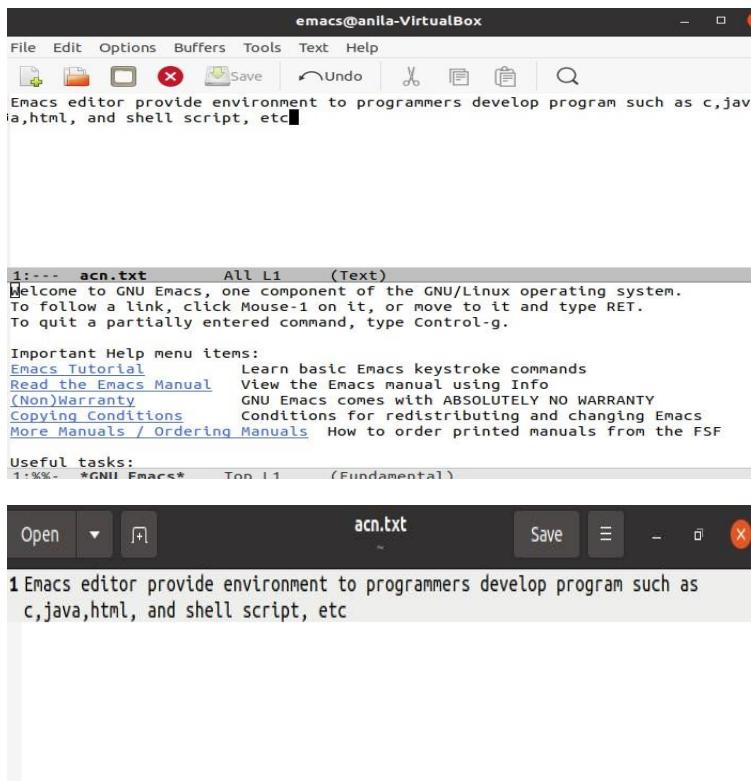
```
[sudo] password for anila:
sudo: 1 incorrect password attempt
anila@anila-VirtualBox:~$ sudo apt-get install emacs
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts
  imagemagick-6-common liblqr-1-0 libm17n-0 libmagickcore-6.q16-6
  libmagickwand-6.q16-6 libotf0 libwebpnu3 m17n-db
Suggested packages:
  mailutils emacs-common-non-dfsg ncurses-term m17n-docs
  libmagickcore-6.q16-6-extra gawk
The following NEW packages will be installed:
  emacs emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts
  imagemagick-6-common liblqr-1-0 libm17n-0 libmagickcore-6.q16-6
  libmagickwand-6.q16-6 libotf0 m17n-db
The following packages will be upgraded:
  libwebpnu3
1 upgraded, 13 newly installed, 0 to remove and 310 not upgraded.
Need to get 39.6 MB of archives.
After this operation, 151 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Check version using the command

\$ emacs --version

```
anila@anila-VirtualBox:~$ emacs --version
GNU Emacs 26.3
Copyright (C) 2019 Free Software Foundation, Inc.
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
You may redistribute copies of GNU Emacs
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.
anila@anila-VirtualBox:~$
```



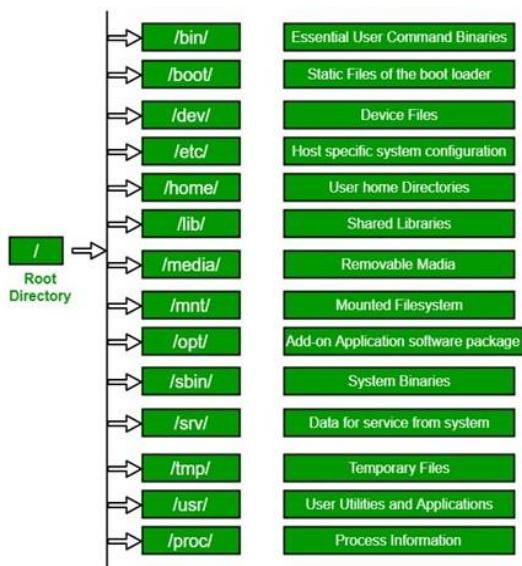


EXPERIMENT- 12

LINUX FILE SYSTEM

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.

- Ø In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Ø Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.
- Ø 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.



ROOT DIRECTORY

All the directories in the Linux system comes under the root directory which is represented by a forward slash (/).

Everything on your Linux system is located under the / directory, known as the root directory. You can think of the / directory as being similar to the C:\ directory on Windows — but this isn't strictly true, as Linux doesn't have drive letters. While another partition would be located at D:\ on Windows, this other partition would appear in another folder under / on Linux.

```
anila@anila-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   sbtn   swapfile  usr
anila@anila-VirtualBox:~$
```

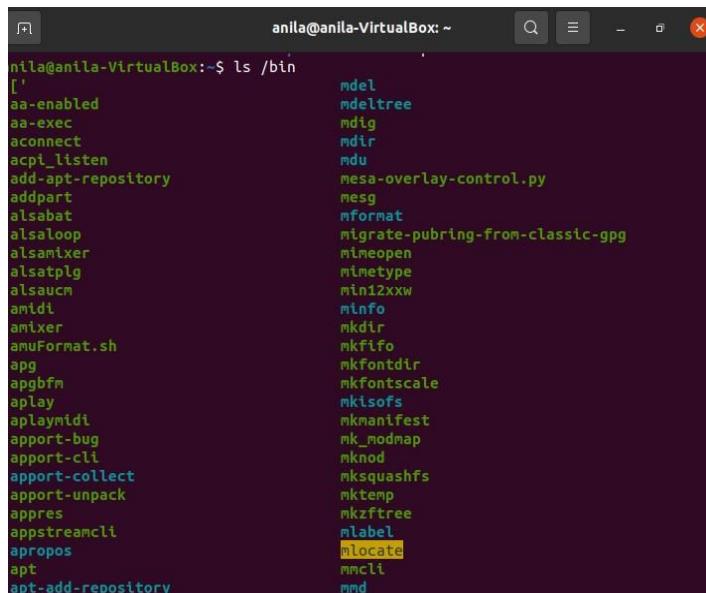
BINARY DIRECTORIES

Binary directory contains following directories:

```
/bin  
/sbin  
/lib  
/opt
```

The /bin directory contains the essential user binaries (programs) that must be present when the system is mounted in single-user mode. Applications such as Firefox are stored in /usr/bin, while important system programs and utilities such as the bash shell are located in /bin. The /usr directory may be stored on another partition — placing these files in the /bin directory ensures the system will have these important utilities even if no other file systems are mounted. The /sbin directory is similar — it contains essential system administration binaries.

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



A screenshot of a terminal window titled "anila@anila-VirtualBox: ~". The command "ls /bin" is run, displaying a long list of binary files. The list includes: aa-enabled, aa-exec, aconnect, acpi_listen, add-apt-repository, addpart, alsabat, alsaloop, alsamixer, alsatplg, alsaucm, amidi, amixer, amuFormat.sh, apg, apgbfm, aplay, aplaymidi, apport-bug, apport-cli, apport-collect, apport-unpack, appres, appstreamcli, apropos, apt, apt-add-repository, md5, mdetree, mdig, mdir, mdu, mesa-overlay-control.py, mesg, mformat, migrate-pubring-from-classic-gpg, mimeopen, mimetyp, mini2xxw, minfo, mkdir, mkfifo, mkfontdir, mkfontscale, mkisofs, mkmanifest, mk_modmap, mknod, mksquashfs, mkttemp, mkzftree, mlabel, mlocate, mmccli, and mmd.

```
anila@anila-VirtualBox:~$ ls /bin
['aa-enabled', 'aa-exec', 'aconnect', 'acpi_listen', 'add-apt-repository', 'addpart', 'alsabat', 'alsaloop', 'alsamixer', 'alsatplg', 'alsaucm', 'amidi', 'amixer', 'amuFormat.sh', 'apg', 'apgbfm', 'aplay', 'aplaymidi', 'apport-bug', 'apport-cli', 'apport-collect', 'apport-unpack', 'appres', 'appstreamcli', 'apropos', 'apt', 'apt-add-repository', 'md5', 'mdetree', 'mdig', 'mdir', 'mdu', 'mesa-overlay-control.py', 'mesg', 'mformat', 'migrate-pubring-from-classic-gpg', 'mimeopen', 'mimetyp', 'mini2xxw', 'minfo', 'mkdir', 'mkfifo', 'mkfontdir', 'mkfontscale', 'mkisofs', 'mkmanifest', 'mk_modmap', 'mknod', 'mksquashfs', 'mkttemp', 'mkzftree', 'mlabel', 'mlocate', 'mmccli', 'mmd']
```

/sbin-'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 purpose.

e.g. fsck, root, init, ifconfig, etc.

```

anila@anila-VirtualBox:~$ ls /sbin
aa-remove-unknown      getweb          pam_exrausers_chkpwd
aa-status              gnome-nenus-blacklist  pam_exrausers_update
aa-teardown            groupadd        pam_getenv
accessdb               groupdel        pam_tally
acpid                  groupmems       pam_timestamp_check
addgnupghome           groupmod        parted
addgroup               grpck          paperconfig
add-shell              grpconv         partprobe
adduser                grpunconv      pccardctl
agetty                 grub-bios-setup pivot_root
alsa                   grub-install    plymouthd
alsabat-test            grub-macless   popcon-largest-unused
alsactl                grub-mkconfig  popularity-contest
alsa-info              grub-mkdevicemap
anacron                grub-probe     poweroff
apparmor_parser        grub-reboot    pppd
apparmor_status        grub-set-default pppdump
aplygnupgdefaults     halt           pppoe-discovery
aptd                  hdparm          pppstats
arpd                  hwclock        pptp
arptables             iconvconfig   pptpsetup
arptables-nft          iio-sensor-proxy pwck
arptables-nft-restore  init           pwconv
arptables-nft-save    insmod         pwunconv
arptables-restore      installkernel raw
arptables-save         install-sgmlcatalog readprofile
aspell-autobuildhash  invoke-rc.d  reboot
avahi-autoind          in             readbignum

```

/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*.

```

anila@anila-VirtualBox:~$ ls /lib
accountsservice        linux-sound-base
apg                     locale
apparmor                lp_solve
apt                     lsb
aspell                 man-db
binfmt.d               memtest86+
bluetooth              mime
bolt                   modprobe.d
brltty                 modules
cnf-update-db           modules-load.d
command-not-found      netplan
console-setup           networkd-dispatcher
cpp                     NetworkManager
crda                   nvidia
cups                   openssh
dbus-1.0               open-vm-tools
debug                  os-prober
dpkg                   os-probes
eject                  os-release
emacs                  packagekit
emacsclient-common     pmciautils
environment.d           pkgconfig
evolution-data-server  pkg-config.multiarch
file                   pm-utils
firefox                policykit-1
firefox-addons          pppd
firmware               pulse-13.99.1
gcc                    python2.7

```

/opt- 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.

The /opt directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the /opt/bin directory and libraries in the /opt/lib directory.

A slight digression: another place where applications and libraries end up in is

/usr/local, When software gets installed here, there will also be /usr/local/bin and /usr/local/lib directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

```
anila@anila-VirtualBox:~$ ls /opt  
anila@anila-VirtualBox:~$
```

CONFIGURATION DIRECTORY

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

Configuration directory have following sub-directories:

/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.

```
anila@anila-VirtualBox:~$ ls /boot  
config-5.11.0-34-generic      memtest86+.elf  
config-5.8.0-43-generic      memtest86+_multiboot.bin  
efi                           System.map-5.11.0-34-generic  
grub                          System.map-5.8.0-43-generic  
initrd.img                     vmlinuz  
initrd.img-5.11.0-34-generic  vmlinuz-5.11.0-34-generic  
initrd.img-5.8.0-43-generic  vmlinuz-5.8.0-43-generic  
initrd.img.old                 vmlinuz.old  
memtest86+.bin                vmlinuz.old  
anila@anila-VirtualBox:~$
```

❖ /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 contain startup and shutdown shell script which is 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 of .conf.

```
anila@anila-VirtualBox:~$ ls /etc/*.conf  
/etc/adduser.conf           /etc/host.conf        /etc/pnm2ppa.conf  
/etc/apg.conf               /etc/kernel-img.conf /etc/popularity-contest.conf  
/etc/appstream.conf         /etc/kerneeloops.conf /etc/resolv.conf  
/etc/brltty.conf            /etc/ld.so.conf       /etc/rsyslog.conf  
/etc/ca-certificates.conf   /etc/libao.conf       /etc/rygel.conf  
/etc/debconf.conf           /etc/libaudit.conf    /etc/sensors3.conf  
/etc/deluser.conf            /etc/logrotate.conf  /etc/sysctl.conf  
/etc/e2scrub.conf           /etc/ltrace.conf     /etc/ucf.conf  
/etc/fprintd.conf           /etc/mke2fs.conf     /etc/updatedb.conf  
/etc/fuse.conf               /etc/mtools.conf    /etc/usb_modeswitch.conf  
/etc/gai.conf                /etc/nsswitch.conf   /etc/xattr.conf  
/etc/hdparm.conf             /etc/pan.conf       anila@anila-VirtualBox:~$
```

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/sssit'.

Inside this directory we have our sub-directories like Desktop, Downloads,

Documents, pictures, etc.

```
anila@anila-VirtualBox:~$ ls /home
anila
anila@anila-VirtualBox:~$ ls /home/anila
acnlab.txt  Desktop   Downloads  Music      new.txt  Public    Videos
acn.txt     Documents  hai.txt   Networks  Pictures  Templates
anila@anila-VirtualBox:~$
```

❖ **/root**

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

'/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.

❖ **/tmp**

The term 'tmp' stands for temporary.

Data stored in '/tmp' is temporary and may use either disk space or RAM.

❖ /dev

/dev contains device files. Many of these are generated at boot time or even on the fly. For example, if you plug in a new webcam or a USB pendrive into your machine, a new device entry will automatically pop up here.

```
anila@anila-VirtualBox:~$ ls /dev
autofs      loop3      stderr  tty36  ttysize   uhid
block       loop4      stdin   tty37  ttys0     uinput
bsg        loop5      stdout  tty38  ttys1     urandom
btrfs-control loop6      tty    tty39  ttys10    userio
bus        loop7      ttys0  ttys1  ttys11    vboxguest
cdrom      loop8      ttys1  ttys10 ttys12    vboxuser
char       loop9      ttys1  ttys11 ttys13    vcs
console    loop-control ttys11 ttys12 ttys14    vcs1
core        mapper    ttys12 ttys13 ttys15    vcs2
cpu_dma_latency mcelog   ttys13 ttys14 ttys16    vcs3
cuse        mem       ttys14 ttys15 ttys17    vcs4
disk       mqueue   ttys15 ttys16 ttys18    vcs5
dma_heap    net       ttys16 ttys17 ttys19    vcs6
dri         null     ttys17 ttys18 ttys2     vcsa
dvd        nvram    ttys18 ttys19 ttys20    vcsa1
cryptifs   port     ttys19 ttys21 ttys21    vcsa2
fb0        ppp      ttys2  ttys20 ttys22    vcsa3
fd        psaux   ttys20 ttys21 ttys23    vcsa4
full       ptmx    ttys21 ttys22 ttys24    vcsa5
fuse        pts     ttys22 ttys23 ttys25    vcsa6
hidraw0   random   ttys23 ttys24 ttys26    vcsu
hpet       rfkill   ttys24 ttys25 ttys27    vcsu1
hugepages  rtc      ttys25 ttys26 ttys28    vcsu2
hwng       rtc0     ttys26 ttys27 ttys29    vcsu3
i2c-0      sda     ttys27 ttys28 ttys3     vcsu4
initctl   sda1    ttys28 ttys29 ttys30    vcsu5
input      sda2    ttys29 ttys30 ttys31    vcsu6
kmsg      sda5     ttys3  ttys31 ttys34    vfs0
```

❖ /mnt

The /mnt directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

❖ /proc

/proc, like /dev is a virtual directory. It contains information about your computer, such as information about your CPU and the kernel your Linux system is running. As with /dev, the files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

```
anila@anila-VirtualBox:~$ ls /proc
1  1314 1619 23 32 6      bus      misc
10 1342 1622 24 339 644   cgroups  modules
100 14 1628 25 4 656    cmdline  mounts
101 1411 1629 253 4178 666  consoles mtrr
103 1428 1630 254 4234 697  cpuinfo  net
104 1442 1632 255 4354 723  crypto   pagetypeinfo
11  1447 1634 256 4355 727  devices   partitions
114 146 1636 257 4360 729  diskstats pressure
117 1468 1641 258 4384 79  dma      sched_debug
118 1475 1646 259 4393 80  driver   schedstat
1181 1488 1647 26 4396 81  dynamic_debug scsi
112 15 165 260 4400 813  execdomains self
1220 1517 1651 261 516 82  fb       slabinfo
123 1521 1657 262 517 83  filesystems softirqs
1233 1522 167 264 550 84  fs       stat
1234 1526 1670 267 551 85  interrupts swaps
1239 1528 1671 268 553 86  iomem   sys
1241 1538 1684 27 554 87  ioports  sysrq-trigger
1244 1542 1694 271 555 89  irq     sysvipc
1248 1545 17 274 556 9   kallsyms thread-self
1250 1555 18 277 557 93  kcore   timer_list
1257 1563 1812 278 563 931  keys    tty
1276 1575 1820 279 564 94  key-users uptime
1281 1580 1828 28 571 96  kmsg   version
1283 1595 188 283 575 97  kpagegroup version_signature
1287 16 189 284 587 98  kpagecount vmallocinfo
1291 1601 19 285 588 99  kpageflags vmsstat
1295 1613 2 29 589  acpi  loadavg zoneinfo
```

❖ /run

/run is another new directory. System processes use it to store temporary data for their own nefarious reasons.

```
anila@anila-VirtualBox:~$ ls /run
acpid.pid    cups      lock      sendsigs.omit.d  thermald
acpid.socket  dbus      log       shm               tmpfiles.d
alsa          fsck      mount     snapd-snap.socket udev
avahi-daemon  gdm3     NetworkManager snapd.socket    udisks2
blkid         gdm3.pid  openvpn   speech-dispatcher user
console-setup initctl   openvpn-client spice-vdagentd utmp
crond.pid     initramfs openvpn-server sudo            uuid
crond.reboot  irqbalance plymouth  systemd
anila@anila-VirtualBox:~$
```

❖ /usr

The /usr directory was where users' home directories were originally kept back in the early days of UNIX. However, now /home is where users kept their stuff as we saw above. These days, /usr contains a mish-mash of directories which in turn

contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services.

You will also find bin, sbin and lib directories in /usr. What is the difference with their root-hanging cousins? Not much nowadays. Originally, the /bin directory (hanging off of root) would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system. /usr/bin on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps.

But many modern Linux distributions just put everything into /usr/bin and have /bin point to /usr/bin just in case erasing it completely would break something. So, while Debian, Ubuntu and Mint still keep /bin and /usr/bin (and /sbin and /usr/sbin) separate; others, like Arch and its derivatives just have one “real” directory for binaries, /usr/bin, and the rest or *bins are “fake” directories that point to /usr/bin.

```
anila@anila-VirtualBox:~$ ls /usr
bin      include  lib32  libexec  local  share
games   lib       lib64  libx32  sbin   src
```

❖ /sys

/sys is another virtual directory like /proc and /dev and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the

/sys/devices/pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel_backlight/brightness file (on

your machine you will probably have a different file). But to do that you have to become a superuser. The reason for that is, as with so many other virtual directories, messing with the

contents and files in /sys can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

```
anila@anila-VirtualBox:~$ ls /sys
block  class  devices  fs      kernel  power
bus    dev    firmware  hypervisor  module
```

❖ /tmp

/tmp contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn't need right now, but may need later on.

You can also use /tmp to store your own temporary files — /tmp is one of the few directories hanging off / that you can actually interact with without becoming a superuser.

❖ /var

/var was originally given its name because its contents was deemed variable, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above.

Be that as it may, /var contains things like logs in the /var/log subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in /var/log; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains spools for tasks. These “tasks” can be the jobs you send to a shared printer when you have to wait because another user is printing a long document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven't mentioned above. In the screenshot, for example, there is a /snap directory. That's because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated snap packages as a way of distributing software. The /snap directory contains all the files and the software installed from snaps.

/var/log/syslog contains lot of system related logfiles.

```
anila@anila-VirtualBox:~$ ls /var
backups  crash  local  log  metrics  run  spool
cache    lib    lock   mail  opt     snap  tmp
anila@anila-VirtualBox:~$
```

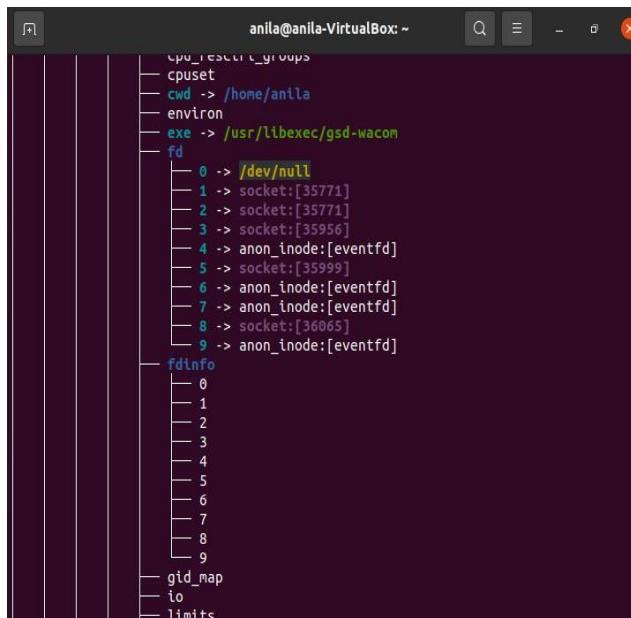
Installing tree

Using sudo apt install tree

```
anila@anila-VirtualBox:~$ sudo apt install tree
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 310 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 0s (87.9 kB/s)
Selecting previously unselected package tree.
(Reading database ... 189842 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
anila@anila-VirtualBox:~$
```

```
$tree /
```

The / in the instruction above refers to the root directory. The root directory is the one from which all other directories branch off from. When you run tree and tell it to start with /, you will see the whole directory tree, all directories and all the subdirectories in the whole system, with all their files, fly by.

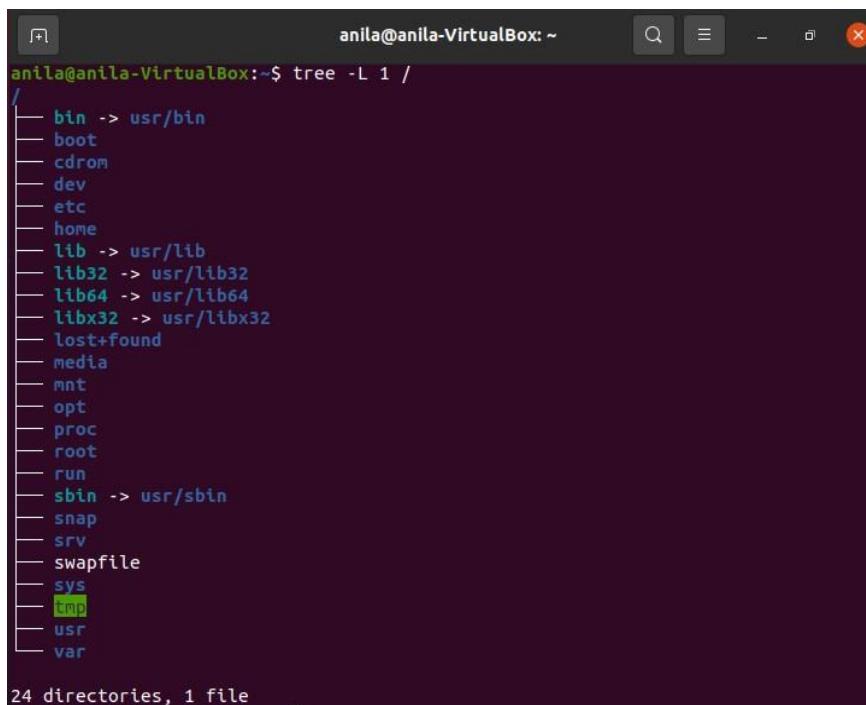


```
anila@anila-VirtualBox: ~
cpuacct_cgroups
cpuset
 cwd -> /home/anila
environ
exe -> /usr/libexec/gsd-wacom
fd
  0 -> /dev/null
  1 -> socket:[35771]
  2 -> socket:[35771]
  3 -> socket:[35956]
  4 -> anon_inode:[eventfd]
  5 -> socket:[35999]
  6 -> anon_inode:[eventfd]
  7 -> anon_inode:[eventfd]
  8 -> socket:[36065]
  9 -> anon_inode:[eventfd]
fdinfo
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
gid_map
io
limits
```

If you have been using your system for some time, this may take a while, because, even if you haven't generated many files yourself, a Linux system and its apps are always logging, caching, and storing temporary files. The number of entries in the file system can grow quite quickly.

Instead, try this:

```
tree -L 1 /
```



```
anila@anila-VirtualBox: ~$ tree -L 1 /
/
 bin -> usr/bin
 boot
 cdrom
 dev
 etc
 home
 lib -> usr/lib
 lib32 -> usr/lib32
 lib64 -> usr/lib64
 libx32 -> usr/libx32
 lost+found
 media
 mnt
 opt
 proc
 root
 run
 sbin -> usr/sbin
 snap
 srv
 swapfile
 sys
 tmp
 usr
 var

24 directories, 1 file
```

CYCLE- 3

EXPERIMENT- 13

FAMILIARISATION TO LINUX SHELL AND SHELL SCRIPTING

Shell Scripting is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime which, reduces programming efforts.

What is shell?

Shell is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program's output in a shell script.

An Operating is made of many components, but its two prime components are –

- Kernel
- Shell

A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.

A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.

Types of Shell

There are two main shells in Linux:

1. **The Bourne Shell:** The prompt for this shell is \$ and its derivatives are listed below:
 - POSIX shell also is known as sh
 - Korn Shell also knew as sh
 - Bourne Again SHell also knew as bash (most popular)
2. **The C shell:** The prompt for this shell is %, and its subcategories are:
 - C shell also is known as csh
 - Tops C shell also is known as tcsh

Creating a shell script

Shell Scripts are written using text editors. On your Linux system, open a text editor program, open a new file to begin typing a shell script or shell programming, then give the shell permission to execute your shell script and put your script at the location from where the shell can find it.

Let us understand the steps in creating a Shell Script:

1. Create a file using a vi editor(or any other editor). Name script file with extension .sh
2. Start the script with #! /bin/sh
3. Write some code.
4. Save the script file as filename.sh
5. For executing the script type bash filename.sh

"#!" is an operator called shebang which directs the script to the interpreter location. So, if we use "#! /bin/sh" the script gets directed to the bourne-shell.

Let's create a small script -

```
#!/bin/sh ls
```

Adding shell comments

Commenting is important in any program. In Shell programming, the syntax to add a comment is ,

```
#comment
```

Shell variables

Variables store data in the form of characters and numbers. Similarly, Shell variables are used to store information and they can be used by the shell only.

Two types of variables

- System variable- Created and maintained by linux itself. This type of variable defined in capital letters.
- Used Defined Variables(UDV)- Created and maintained by user. This type of variable defined in small letters.

Read statement

Use to get input (data from user) from keyboard and store (data) to variable. Syntax:

- read variable1, variable2,...variableN

Conditional statements

if condition

- Used for making the decisions in shell script
- If the condition is true , Command1 is executed.

Syntax:

if condition

then

command1 if condition is true or if exit status of condition is 0

fi

if else fi

- If the condition is true
- Command 1 is executed
- Otherwise, Command 2 is executed
- fi represents loop termination.

Syntax

if [expression]

then

Statement1

else

Statement2

fi

if elif else fi

- To use multiple conditions in one if-else block, then elif keyword is used in shell.

Syntax

if [expression 1]

then

Statement1

Statement2

elif [expression2]

then

else

Statement3

Statement4

fi

Statement5

- If expression1 is true then it executes statement 1 and 2, and this process continues.
- If none of the condition is true then it processes else part.

Loops in Shell scripting

Bash supports two types of loop

- for loop
- while loop

Note that each and every loop

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

for loop Syntax

for (exp1;exp2;exp3)

do

repeat all statements between do and done until exp2 is true

done

While loop

Syntax

while [condition]

do

command1

command2

...

Done

EXPERIMENT-14

LINUX SHELL SCRIPTING PROBLEMS

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

```
#!/bin/bash
echo "Hello, $LOGNAME"
echo "Current date is $(date)"
echo "User is $(whoami)"
echo "Current directory $(pwd)"
```

```
Hello, anila
Current date is Tuesday 21 September 2021 01:23:12 AM IST
User is anila
Current directory /home/anila/Networks
```

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

```
cat etc/passwd
```

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60: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-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
```

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

We can write a bash script that can accept arguments from the command line in the following manner.

```
$ ./scriptName "arg1" "arg2"..."argn"
#!/bin/bash
# sh parameters .sh 50,51,52
echo "First parameter is 50"
echo "Second parameter is 51"
echo "Third parameter is 52"
exit 0
```

```
First parameter is 50
Second parameter is 51
Third parameter is 52
```

4. How to set an array in Linux?

1. Indirect Declaration

In Indirect declaration, we assigned a value in a particular index of Array Variable. No need to first declare.

ARRAY NAME[INDEXNR]=value

2. Explicit Declaration

In Explicit Declaration, First We declare an array then assign the values.

declare -a ARRAYNAME

3. Compound Assignment

In Compound Assignment, We declare an array with a bunch of values.

We can add other values later too.

ARRAYNAME=(value1 value2 valueN)

5. How to check if a directory exists?

```
if [ -d "/path/to/dir" ]
```

```
then
```

```
    echo "Directory /path/to/dir exists."
```

```
else
```

```
    echo "Error: Directory /path/to/dir does not exists."
```

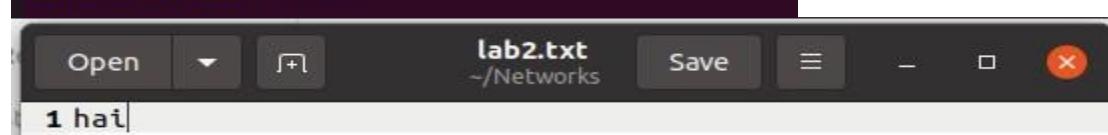
```
fi
```

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?

```
sed -i 's/old-text/new-text/g' input.txt
```



8. Write a script to compare numbers?

```
#!/bin/bash

# Script to do numeric comparisons

var1=10

var2=20

if [ $var2 -gt $var1 ] then

echo "$var2 is greater than $var1"

fi

# Second comparison

If [ $var1 -gt 30] then

echo "$var is greater than 30"

else echo "$var1 is less than 30"

fi
```

```
Enter the first number:12
Enter the Second number:8
8 < 12
```

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””?

```
#!/bin/bash FILE=""

if [ -e "$FILE" ]

then

echo "$FILE passwords are enabled"

fi

if [ -x "$FILE" ]

then

echo "You have permission to execute $FILE" else echo "You do Not have permissions to execute $FILE" fi
```

10. How to print all array indexes?

```
echo ${ARRAYNAME[*]}

#!/bin/bash
```

```
# To declare static Array  
arr=(green pink green yellow red) #  
  
To print all elements of array  
echo ${arr[@]}  
echo ${arr[*]}  
echo ${arr[@]:0}  
echo ${arr[*]:0}
```

```
green pink green yellow red  
green pink green yellow red  
green pink green yellow red  
green[*]:0}
```

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

```
#!/bin/bash  
ls -lrt | grep ^- | awk 'END{print $NF}'
```

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

running the test script.

```
#!/bin/bash  
dir=$1  
for file in `ls $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.

```
#!/bin/bash  
if [ $# -ne 1 ] then  
    echo "Provide the correct input in the above format."  
    echo "Usage: $0 number"  
    echo "this script will reverse the given number."  
    echo "For eg. $0 1234, will print 4321"  
    exit 1
```

```

fi
n=$1
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"

```

Provide the correct input in the above format,
Usage: reverse.sh number
this script will reverse the given number.
For eg.reverse.sh 1234, will print 4321

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

In Linux or Unix-like systems, you may come across file names including the following special characters, white spaces, backslashes and more.

-
--
;
&
\$?
*

Bash shell considers most of the above special characters as commands. Thus, the “rm” command may not be able to delete such files. The simplest way to delete files having special characters in its name is by using the inode number

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

```

$ cat animal.txt
tiger bear elephant
tiger bear

```

bear

Following test script/command will count the unique words.

```
$ awk '{for(i=1;i<=NF;i++)a[$i]++;}END{for(i in a){print i, a[i];}}' animal.txt
```

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

```
# Program for Fibonacci  
  
# Series  
  
# Static input fo N  
  
N=6  
  
# First Number of the  
  
# Fibonacci Series  
  
a=0  
  
# Second Number of the # Fibonacci Series  
  
b=1  
  
echo "The Fibonacci series is : "  
  
for (( i=0; i<N; i++ )) do  
  
    echo -n "$a "  
  
    fn=$((a + b))  
  
    a=$b  
  
    b=$fn  
  
done  
  
# End of for loop
```

```
The fibonacci series is:  
0  
1  
1  
2  
3  
5
```

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.

The following is the test script/command which recursively searches for the “.txt” files and returns the total occurrences of a word <Linux>.

```
$ find . -name *.txt -exec grep -c Linux '{}' \; | awk '{x+=$0;}END{print x}'
```

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 the 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.

Password Validation Script

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

CYCLE- 4

EXPERIMENT- 15

INSTALLATION OF LAMP STACK

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

Step 1 — Installing Apache and Updating the Firewall

```
sudo apt update
```

```
[sudo] password for anila:  
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease  
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]  
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Get:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [538 kB]  
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,222 kB]  
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [874 kB]  
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [261 kB]  
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metad ata [282 kB]  
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metad ata [14.3 kB]  
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packa ges [454 kB]  
Get:12 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted Translation -en [65.0 kB]
```

```
sudo apt install apache2
```

```
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi  
  libgstreamer-plugins-bad1.0-0 libva-wayland2  
Use 'sudo apt autoremove' to remove them.  
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 315 not upgraded.
```

```
sudo ufw app list
```

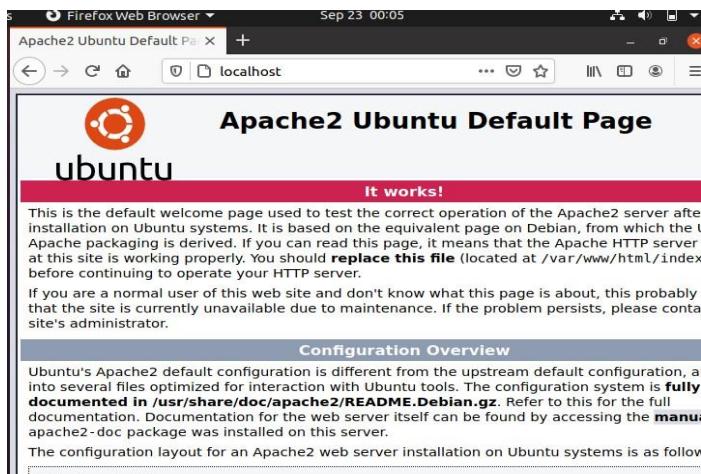
```
Available applications:  
  Apache  
  Apache Full  
  Apache Secure  
  CUPS
```

```
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
```

You will see the default Ubuntu 18.04 Apache web page, which is there for informational and testing purposes. It should look something like this:



```
anila@anila-VirtualBox:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libcurl4
The following NEW packages will be installed:
curl
The following packages will be upgraded:
libcurl4
1 upgraded, 1 newly installed, 0 to remove and 314 not upgraded.
Need to get 396 kB of archives.
After this operation, 413 kB of additional disk space will be used.
```

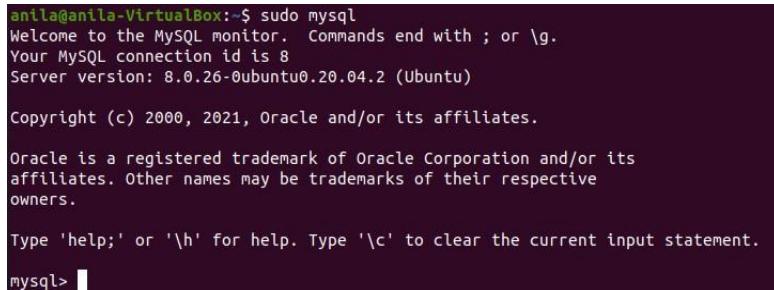
Step 2 — Installing MySQL

```
sudo apt install mysql-server
```

```
anila@anila-VirtualBox:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
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 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
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, 16 newly installed, 0 to remove and 314 not upgraded.
```

```
sudo mysql_secure_installation
```

```
sudo mysql
```



```
anila@anila-VirtualBox:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

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

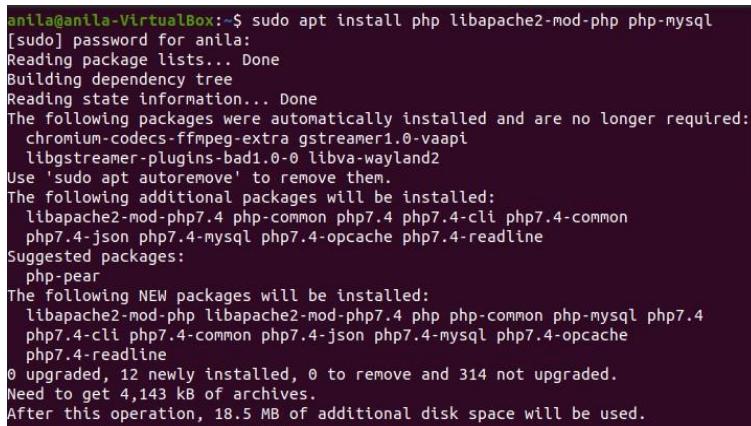
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> 
```

Step 3 — Installing PHP sudo apt install php

```
libapache2-mod-php php-mysql
```



```
anila@anila-VirtualBox:~$ sudo apt install php libapache2-mod-php php-mysql
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
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 314 not upgraded.
Need to get 4,143 kB of archives.
After this operation, 18.5 MB of additional disk space will be used.
```

Step 4 — Testing PHP Processing on your Web Server

In order to test that your system is properly configured for PHP, create a PHP script called info.php. In order for Apache to find this file and serve it correctly, it must be saved to your web root directory.

Create the file at the web root you created in the previous step by running:

```
$ sudo nano /var/www/your_domain/info.php
```

This will open a blank file. Add the following text, which is valid PHP code, inside the file:

```
<?php
phpinfo();
```

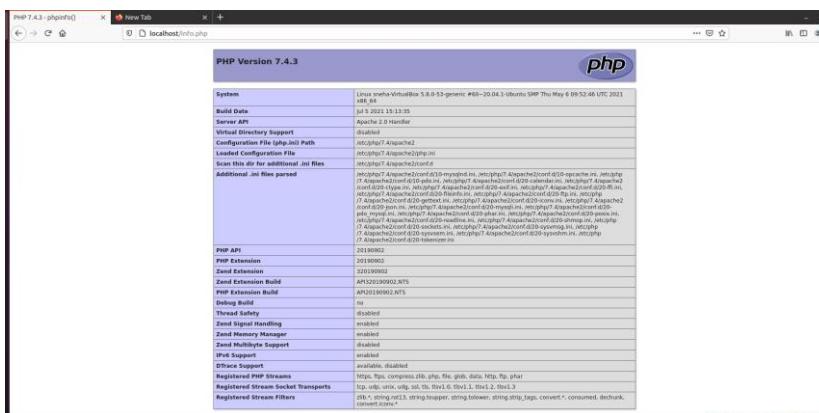
```
GNU nano 4.8          /var/www/html/info.php          Modified
<?php
phpinfo();
?>

File Name To Write: /var/www/html/info.php
~G Get Help      M-D DOS Format    M-A Append      M-B Backup File
~C Cancel        M-M Mac Format     M-P Prepend    M-T To Files
```

When you are finished, save and close the file. Now you can test whether your web server is able to correctly display content generated by this PHP script. To try this out, visit this page in your web browser. You'll need your server's public IP address or domain name again.

The address you will want to visit is:

http://your_domain/info.php



EXPERIMENT- 16

INSTALLATION OF 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 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 framework in developing web application.
- It includes namespaces and interfaces, thus helps to organize and manage resources.

Composer

Composer is a tool which includes all the dependencies and libraries. It allows a user to create a project with respect to the mentioned framework (for example, those used in Laravel installation). Third party libraries can be installed easily with help of composer.

All the dependencies are noted in composer.json file which is placed in the source folder.

Artisan

Command line interface used in Laravel is called Artisan. It includes a set of commands which assists in building a web application. These commands are incorporated from Symphony framework, resulting in add-on features in Laravel 5.1 (latest version of Laravel).

Features of Laravel

Laravel offers the following key features which makes it an ideal choice for designing web applications –

Modularity

Laravel provides 20 built in libraries and modules which helps in enhancement of the application. Every module is integrated with Composer dependency manager which eases updates.

Testability

Laravel includes features and helpers which helps in testing through various test cases. This feature helps in maintaining the code as per the requirements.

Routing

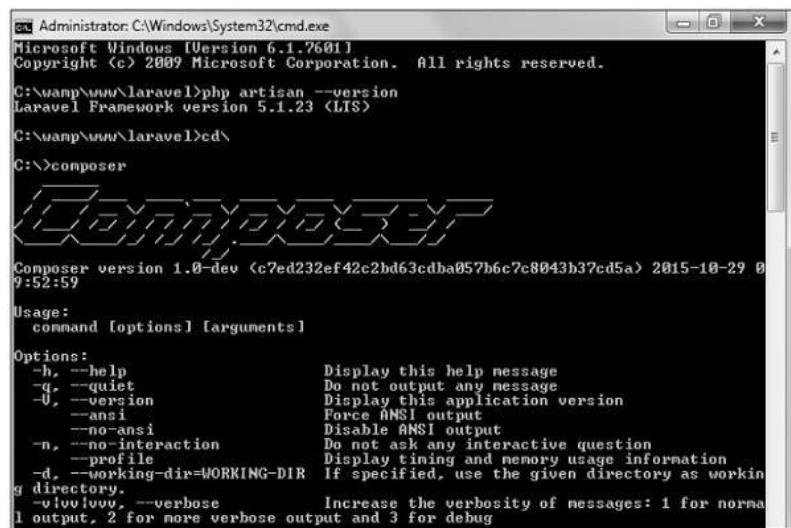
Laravel provides a flexible approach to the user to define routes in the web application. Routing helps to scale the application in a better way and increases its performance.

INSTALLATION:

For managing dependencies, Laravel uses composer. Make sure you have a Composer installed on your system before you install Laravel. In this chapter, you will see the installation process of Laravel.

You will have to follow the steps given below for installing Laravel onto your system –

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



The screenshot shows a Windows command prompt window titled 'Administrator: C:\Windows\System32\cmd.exe'. It displays the output of the following commands:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\wamp\www\laravel>php artisan --version
Laravel Framework version 5.1.23 (LTS)

C:\wamp\www\laravel>cd\

C:\>composer
```

Composer version 1.0-dev (c7ed232ef42c2bd63cdba057b6c7c8043b37cd5a) 2015-10-29 09:52:59

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
-d, --working-dir=WORKING-DIR	If specified, use the given directory as working directory.
--vvv vvvv v	Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

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

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.

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

Now, we will focus on installation of version 5.7. In Laravel version 5.7, you can install the complete framework by typing the following command –

`composer create-project laravel/laravel test dev-develop`

the output of the command is as shown below –

```

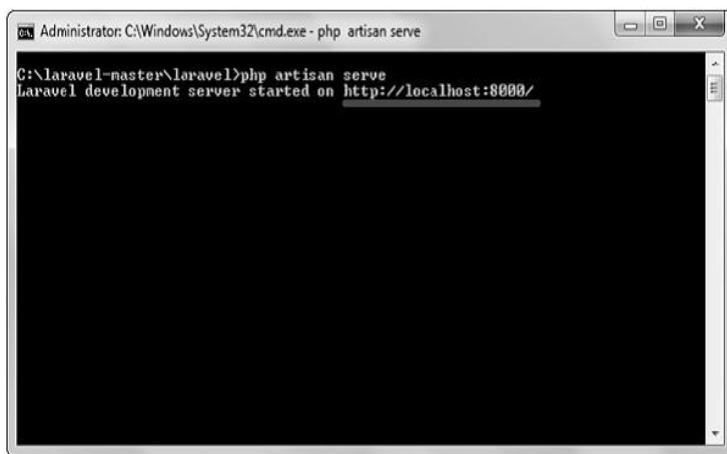
➔ code composer create-project laravel/laravel test dev-develop
Installing laravel/laravel (dev-develop d6acacd21cb2288713d9c09a31f9b4ab86f116039)
  - Installing laravel/laravel (dev-develop develop): Cloning develop from cache
Created project in test
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 71 installs, 0 updates, 0 removals
  - Installing vlucas/phpdotenv (v2.5.1): Loading from cache
  - Installing symfony/css-selector (v4.1.3): Loading from cache
  - Installing titjerkverkoyen/css-to-inline-styles (2.2.1): Loading from cache
  - Installing symfony/polyfill-php72 (v1.9.0): Loading from cache
  - Installing symfony/polyfill-mbstring (v1.9.0): Loading from cache
  - Installing symfony/var-dumper (v4.1.3): Loading from cache
  - Installing symfony/routing (v4.1.3): Loading from cache
  - Installing symfony/process (v4.1.3): Loading from cache
  - Installing symfony/polyfill ctype (v1.9.0): Loading from cache
  - Installing symfony/http-foundation (v4.1.3): Loading from cache
  - Installing symfony/event-dispatcher (v4.1.3): Loading from cache
  - Installing psr/log (1.0.2): Loading from cache
  - Installing symfony/debug (v4.1.3): Loading from cache
  - Installing symfony/http-kernel (v4.1.3): Loading from cache
  - Installing paragonie/random_compat (v9.99.99): Loading from cache

```

The Laravel framework can be directly installed with develop branch which includes the latest framework.

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 –



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.

CYCLE- 5

EXPERIMENT- 17

INTRODUCTION 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

1. 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:
 1. Network connection can be established
 2. Speed of the connection
- ❖ Ping is used to test a network host's capacity to interact with another host. Just enter the command Ping, followed by the target host's name or IP address.

```
anilla@anilla-VirtualBox:~$ ping 192.162.1.6
PING 192.162.1.6 (192.162.1.6) 56(84) bytes of data.
```

```
anila@anila-VirtualBox:~$ ping google.com
PING google.com (142.250.192.78) 56(84) bytes of data.
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=1 ttl=114 time=82.2 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=2 ttl=114 time=178 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=3 ttl=114 time=202 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=4 ttl=114 time=231 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=5 ttl=114 time=250 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=6 ttl=114 time=73.3 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=7 ttl=114 time=84.3 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=8 ttl=114 time=120 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=9 ttl=114 time=111 ms
64 bytes from bom12s16-in-f14.1e100.net (142.250.192.78): icmp_seq=10 ttl=114 time=111 ms
```

2. 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

```
anila@anila-VirtualBox:~$ hostnamectl
Static hostname: anila-VirtualBox
  Icon name: computer-vm
    Chassis: vm
  Machine ID: a2209d4d8cd14c35874291120f5bf820
    Boot ID: 024faac9e72d4d7ca09e3480508e04fd
  Virtualization: oracle
Operating System: Ubuntu 20.04.2 LTS
      Kernel: Linux 5.11.0-34-generic
    Architecture: x86-64
anila@anila-VirtualBox:~$ hostname
anila-VirtualBox
anila@anila-VirtualBox:~$ hostname -f
anila-VirtualBox
anila@anila-VirtualBox:~$ hostname -i
127.0.1.1
anila@anila-VirtualBox:~$
```

3.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

```

anila@anila-VirtualBox:~$ sudo apt install net-tools
[sudo] password for anila:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 314 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 2s (130 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 191093 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb .

```

```

anila@anila-VirtualBox:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
udp    0      0 anila-VirtualBox:bootpc _gateway:bootps      ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State     I-Node Path
unix  2      [ ]      DGRAM          30470  /run/user/1000/syste
md/notify
unix  4      [ ]      DGRAM          15085  /run/systemd/notify
unix  2      [ ]      DGRAM          15099  /run/systemd/journal
/syslog
unix  16     [ ]      DGRAM          15109  /run/systemd/journal
/dev-log
unix  8      [ ]      DGRAM          15113  /run/systemd/journal
/socket
unix  3      [ ]      STREAM   CONNECTED  35552
unix  2      [ ]      DGRAM          32953
unix  3      [ ]      STREAM   CONNECTED  38486  @/tmp/dbus-JgQiH74Sv
e
unix  3      [ ]      STREAM   CONNECTED  36893
unix  3      [ ]      STREAM   CONNECTED  34339  /run/user/1000/bus
unix  3      [ ]      STREAM   CONNECTED  34133

```

```

anila@anila-VirtualBox:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp    0      0 localhost:mysql           0.0.0.0:*          LISTEN
tcp    0      0 localhost:domain         0.0.0.0:*          LISTEN
tcp    0      0 localhost:ipp            0.0.0.0:*          LISTEN
tcp    0      0 localhost:33060          0.0.0.0:*          LISTEN
tcp6   0      0 [::]:http              [::]:*             LISTEN
tcp6   0      0 ip6-localhost:ipp        [::]:*             LISTEN
anila@anila-VirtualBox:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
udp    0      0 localhost:domain         0.0.0.0:*          ESTABLISHED
udp    0      0 anila-VirtualBox:bootpc _gateway:bootps      ESTABLISHED
udp    0      0 0.0.0.0:mdns           0.0.0.0:*
udp    0      0 0.0.0.0:57834          0.0.0.0:*
udp    0      0 0.0.0.0:631           0.0.0.0:*
udp6   0      0 [::]:mdns             [::]:*             LISTEN
udp6   0      0 [::]:35098            [::]:*             LISTEN
anila@anila-VirtualBox:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp    0      0 localhost:mysql           0.0.0.0:*          LISTEN
tcp    0      0 localhost:domain         0.0.0.0:*          LISTEN
tcp    0      0 localhost:ipp            0.0.0.0:*          LISTEN
tcp    0      0 localhost:33060          0.0.0.0:*          LISTEN
tcp6   0      0 [::]:http              [::]:*             LISTEN

```

```

anila@anila-VirtualBox:~$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp    0      0 localhost:mysql           0.0.0.0:*          LISTEN
tcp    0      0 localhost:domain         0.0.0.0:*          LISTEN
tcp    0      0 localhost:ipp            0.0.0.0:*          LISTEN
tcp    0      0 localhost:33060          0.0.0.0:*          LISTEN
tcp6   0      0 [::]:http              [::]:*             LISTEN
tcp6   0      0 ip6-localhost:ipp        [::]:*             LISTEN
anila@anila-VirtualBox:~$ netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
udp    0      0 localhost:domain         0.0.0.0:*
udp    0      0 0.0.0.0:mdns           0.0.0.0:*
udp    0      0 0.0.0.0:57834          0.0.0.0:*
udp    0      0 0.0.0.0:631           0.0.0.0:*
udp6   0      0 [::]:mdns             [::]:*             LISTEN
udp6   0      0 [::]:35098            [::]:*             LISTEN
anila@anila-VirtualBox:~$ 
```

4. 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.

```
anila@anila-VirtualBox:~$ ifconfig -a
Command 'ifconfig' not found, but can be installed with:
sudo apt install ipmiutil

anila@anila-VirtualBox:~$ sudo apt install ipmiutil
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libtirpc-common libtirpc3 rpcbind
The following NEW packages will be installed:
ipmiutil libtirpc-common libtirpc3 rpcbind
```

```
anila@anila-VirtualBox:~$ ifconfig -s
ipmiutil config ver 3.15
config: option requires an argument -- 's'
Usage: ipmiutil config [-clmpxLNUPREFTJYV -r <file> -s <file>]
where -l  Lists BMC configuration parameters
      -r Restores BMC configuration from <file>
      -s Saves BMC configuration to <file>
      -c canonical output with delimiter '|'
      -m Set BMC MAC during restore
      -x show eXtra debug messages
      -p <psw> specify a user password to set
      -L 3    specify lan channel number 3
      -N node Nodename or IP address of target system
      -U user Username for remote node
      -P/-R pswd Remote Password
      -E   use password from Environment IPMI_PASSWORD
      -F   force driver type (e.g. imb, lan2)
      -J 0 use lanplus cipher suite 0: 0 thru 14, 3=default
      -T 1 use auth Type: 1=MD2, 2=MD5(default), 4=Pswd
      -V 2 use privilege level: 2=user(default), 4=admin
      -Y   prompt for remote password
      -Z   set slave address of local MC
Segmentation fault (core dumped)
anila@anila-VirtualBox:~$
```

5. NSLOOKUP

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific

DNS record. It is also used to troubleshoot DNS related problems.

nslookup google.com :

nslookup followed by the domain name will display the “A Record” (IP Address) of the domain. Use this command to find the address record for a domain. It queries to domain name servers and get the details.

nslookup 192.168.0.10 : Reverse DNS lookup we can also do the reverse DNS look-up by providing the IP Address as argument to 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

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

Non-authoritative answer:
Name:  google.com
Address: 142.250.192.110
Name:  google.com
Address: 2404:6800:4009:82a::200e

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

Authoritative answers can be found from:

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

Non-authoritative answer:
Name:  google.com
Address: 142.250.183.110
Name:  google.com
Address: 2404:6800:4009:823::200e
google.com      nameserver = ns3.google.com.
google.com      rdata 257 = 0 issue "pki_gaaa"
```

```

anila@anila-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 = 2021092000
    refresh = 300
    retry = 180
    expire = 604800
    minimum = 14400

Authoritative answers can be found from:

anila@anila-VirtualBox:~$ nslookup -type = ns google.com
*** Invalid option: type
Usage:
nslookup [-opt ...]      # interactive mode using default server
nslookup [-opt ...] - server # interactive mode using 'server'
nslookup [-opt ...] host    # just look up 'host' using default server
nslookup [-opt ...] host server # just look up 'host' using 'server'
anila@anila-VirtualBox:~$ 

```

6. 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

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

```

anila@anila-VirtualBox:~$ traceroute google.com
traceroute to google.com (142.250.192.110), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.239 ms  37.721 ms  37.653 ms
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *

```

```

anila@anila-VirtualBox:~$ traceroute -m 5 google.com
traceroute to google.com (142.250.192.110), 5 hops max, 60 byte packets
1 _gateway (10.0.2.2)  0.830 ms  0.731 ms  16.716 ms
2 * * *
3 * * *
4 * * *
5 * * *

anila@anila-VirtualBox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         _gateway       0.0.0.0        UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0       255.255.255.0  U      100    0        0 enp0s3
link-local      0.0.0.0       255.255.0.0   U      1000   0        0 enp0s3
anila@anila-VirtualBox:~$ 

```

7.ROUTE route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table.

install net-tools by \$sudo apt-get install net-tools

\$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-18

FAMILIARISATION OF STATIC AND DYNAMIC IP

STATIC AND DYNAMIC IP STATIC IP

A fixed IP address is called static IP address, i.e. it never changes.

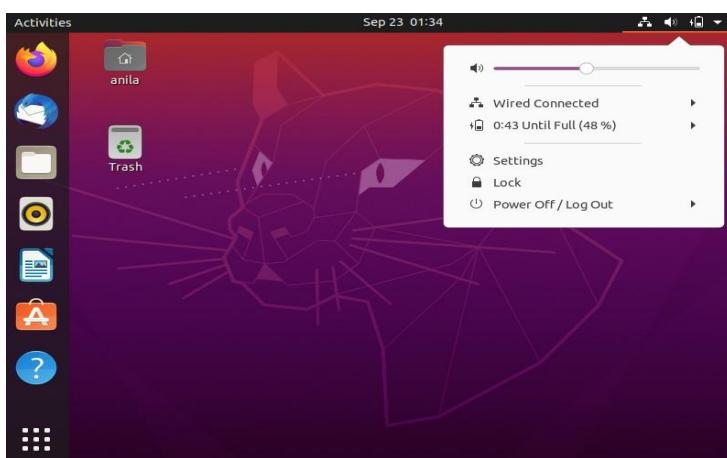
It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.

It is used to connect to an IP camera, home file server, game server, and many other devices.

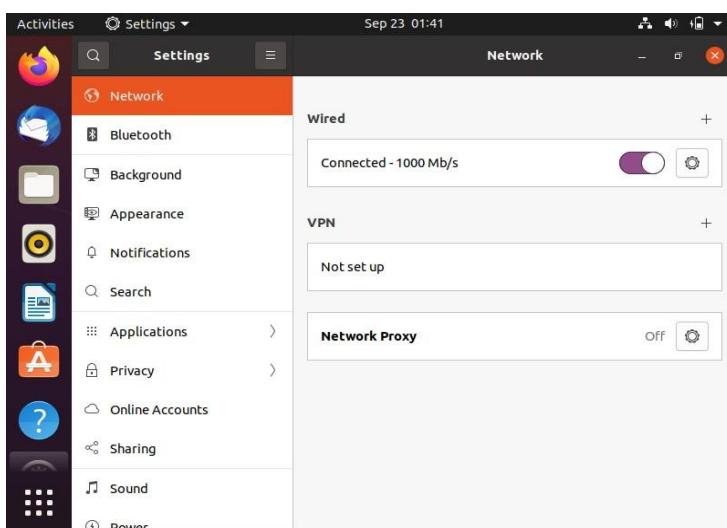
A static IP address is necessary only for servers and not for personal PCs

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.



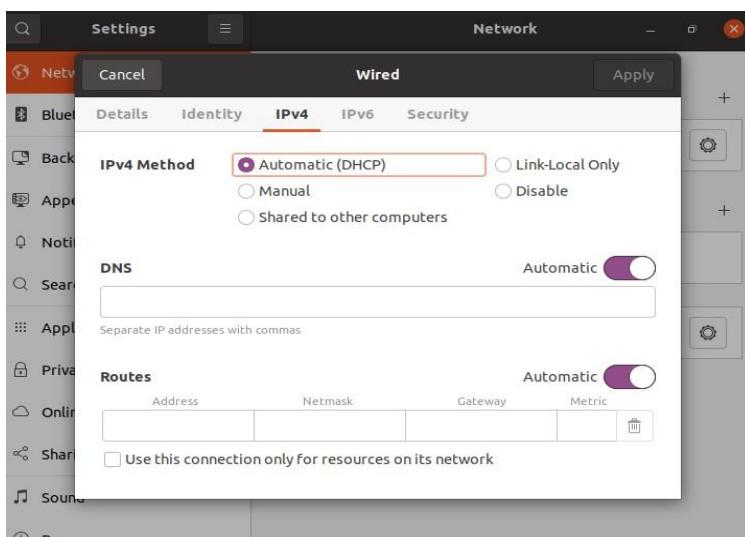
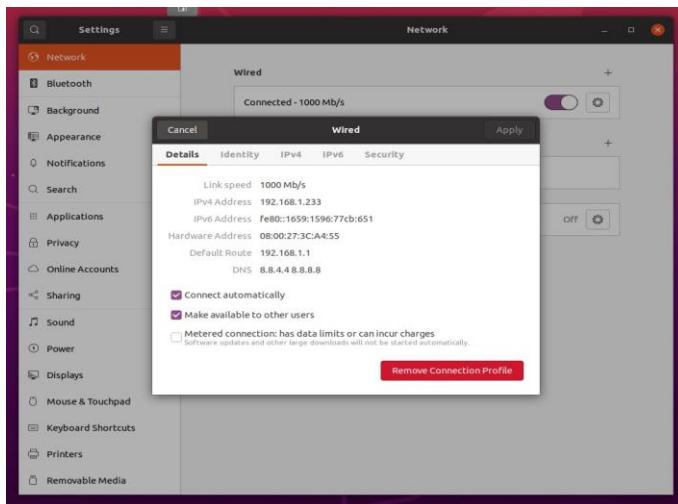
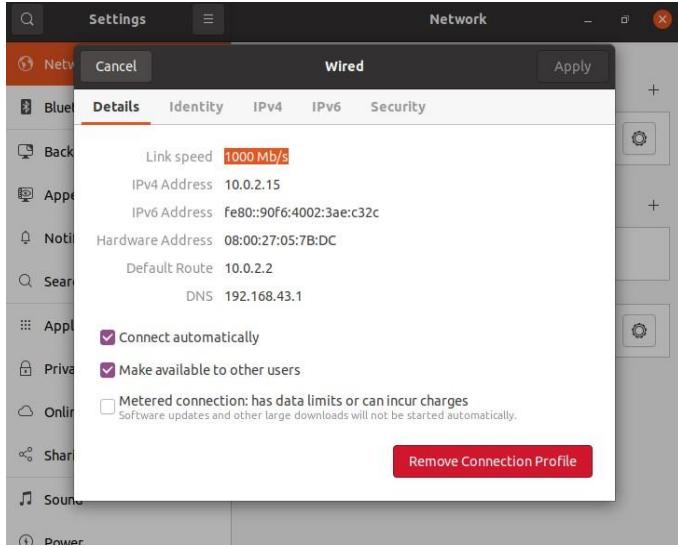
Click on the settings icon to start IP address configuration



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

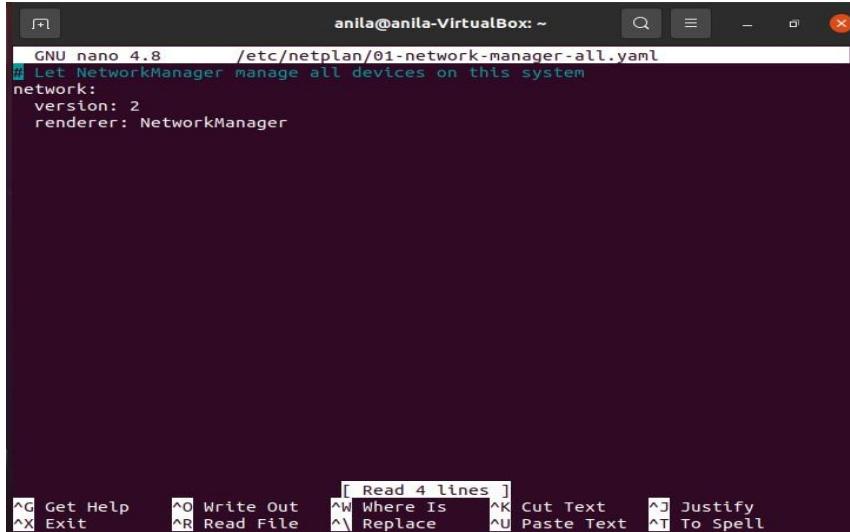
Configuring dynamic ip addresses

Step1: type the command in the terminal

sudo nano /etc/netplan/01-network-manager-all.yaml step2:Now find the name of the network interface you want to configure and insert the following lines:

dhcp4: yes

dhcp6: yes



```
anila@anila-VirtualBox: ~
GNU nano 4.8      /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

The screenshot shows a terminal window with the title "anila@anila-VirtualBox: ~". The command "sudo nano /etc/netplan/01-network-manager-all.yaml" was run. The file content is displayed in the nano text editor. At the bottom of the screen, there is a menu bar with options like "Read 4 lines", "Cut Text", "Justify", etc., and a series of keyboard shortcuts for navigating and editing the text.

Step3:Apply the changes with sudo netplan apply command

EXPERIMENT-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 value 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 subnet mask 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-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:

11000000.10101000.01111011.10000100 - 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:

11000000.10101000.01111011.00000000 - Network address (192.168.123.0)

00000000.00000000.00000000.10000100 - Host address (000.000.000.132)

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 111111.1111111.1111111.11000000 255.255.255.224

1111111.11111111.1111111.11100000

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- 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
```

```
anila@anila-VirtualBox:~$ sudo ufw status
[sudo] password for anila:
Status: inactive
anila@anila-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
anila@anila-VirtualBox:~$ sudo ufw status
Status: active
```

To disable ufw

Step1:To disable firewall

```
sudo ufw disable
```

Step2:to check status

```
sudo ufw status
```

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

EXPERIMENT- 22

WIRESHARK AND TCPDUMP

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 ?

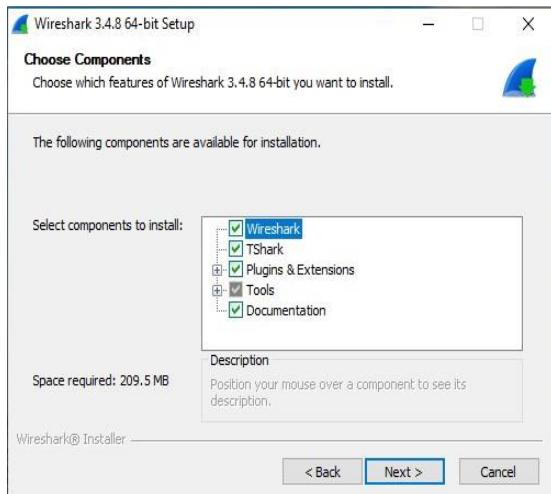
- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- Testers use it to detect defects
- People use it to learn network protocol internals

STEPS TO INSTALL WIRESHARK

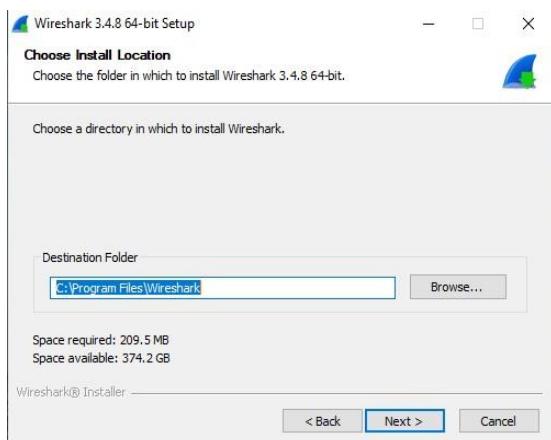
- Step1: Go to www.wireshark.org ->download 64bit package.
- Step 2 : run application and click on noted



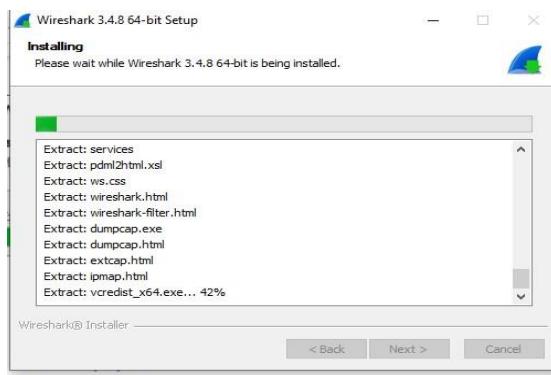
- Step 3: select components and click next



- Step 4: choose default destination location

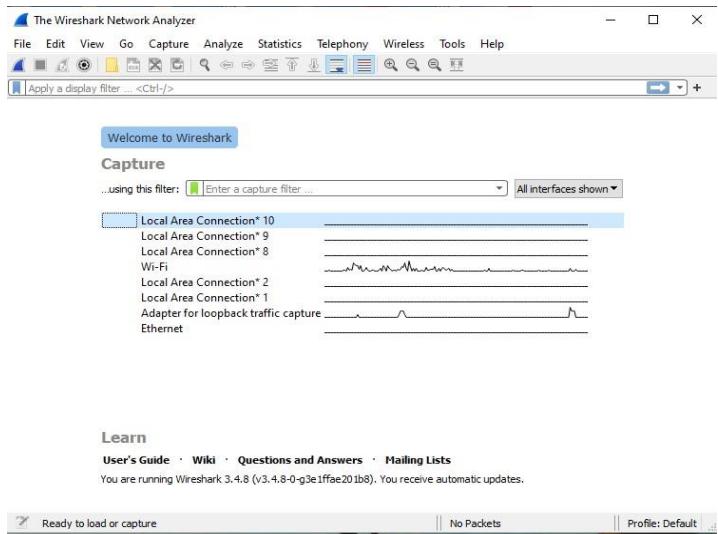


- Step 5: installation start running



- 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
```

```
sudo apt install tcpdump
```

```
anila@anila-VirtualBox:~$ sudo apt install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-4).
tcpdump set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libdouble-conversion3
  libgstreamer-plugins-bad1.0-0 libminizip1 libpcre2-16-0 libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimediasupport5
  libqt5multimediawidgets5 libqt5networks5 libqt5opengl5 libqt5svg5
  libqt5widgets5 libva-wayland2 libxcb-xinerama0 libxcb-xinput0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 314 not upgraded.
anila@anila-VirtualBox:~$
```

tcpdump Command Examples

1. Display Available Interfaces

```
# tcpdump -D
```

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

```
anila@anila-VirtualBox:~$ sudo tcpdump -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
01:13:03.314845 IP6 anila-VirtualBox.mdns > ff02::fb.mdns: 0 [2q] PTR (QM)? _ipp._tcp.local. (45)
01:13:03.315373 IP anila-VirtualBox.mdns > 224.0.0.251.mdns: 0 [2q] PTR (QM)? _ipp._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
01:13:03.385710 IP localhost.42563 > localhost.domain: 30716+ [1au] PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (101)
01:13:03.389350 IP anila-VirtualBox.51081 > 192.168.43.1.domain: 46472+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
01:13:05.316592 IP 192.168.43.1.domain > anila-VirtualBox.51081: 46472 NXDomain 0/1/0 (154)
01:13:05.319673 IP localhost.domain > localhost.42563: 30716 NXDomain 0/0/1 (101)
01:13:08.489872 IP localhost.33016 > localhost.domain: 6144+ [1au] PTR? 53.0.0.127.in-addr.arpa. (52)
01:13:29.339381 IP anila-VirtualBox.44728 > golem.canonical.com.ntp: NTPv4, Client, length 48
```

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

```
anila@anila-VirtualBox:~$ sudo tcpdump -A -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked vi), capture size 262144 bytes
01:16:42.333667 IP anila-VirtualBox.40794 > golem.canonical.com.ntp: NTPv4, Client, length 48
E..L..@.PM
....[.Y..Z.{.8..#.....X"..
01:16:42.336756 IP localhost.45178 > localhost.domain: 53850+ [1au] PTR? 199.89
.189.91.in-addr.arpa. (55)
E..SD..@. ....5.z.5.?...Z. ....199.89.189.91.in-addr.arpa....)....
...
01:16:42.337804 IP localhost.domain > localhost.45178: 53850 1/0/1 PTR golem.ca
nonical.com. (88)
E..t*B@. ....5....5.z.`...Z. ....199.89.189.91.in-addr.arpa.....
.
```

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

```
anila@anila-VirtualBox:~$ sudo tcpdump -c 5 -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked vi), capture size 262144 bytes
01:22:31.282055 IP anila-VirtualBox.60498 > 192.168.43.1.domain: 37395+ AAAA? c
onnectivity-check.ubuntu.com. (47)
01:22:31.288849 IP 192.168.43.1.domain > anila-VirtualBox.60498: 37395 0/0/0 (4
7)
01:22:31.289385 IP localhost.56966 > localhost.domain: 10908+ [1au] PTR? 1.43.1
68.192.in-addr.arpa. (54)
01:22:31.290240 IP anila-VirtualBox.34080 > 192.168.43.1.domain: 6956+ PTR? 1.4
3.168.192.in-addr.arpa. (43)
01:22:31.292859 IP localhost.52797 > localhost.domain: 33245+ [1au] AAAA? conne
ctivity-check.ubuntu.com. (58)
5 packets captured
24 packets received by filter
13 packets dropped by kernel
anila@anila-VirtualBox:~$
```

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

```
anila@anila-VirtualBox:~$ sudo tcpdump -XX -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked vi), capture size 262144 bytes
01:24:00.585249 IP anila-VirtualBox.42929 > 192.168.43.1.domain: 1615+ A? conne
ctivity-check.ubuntu.com. (47)
    0x0000:  0004 0001 0006 0800 2705 7bdc 0000 0800 .....'{.....
    0x0010:  4500 004b 88d8 4000 4011 ba5c 0a00 020f E..K..@.0.0.0.
    0x0020:  c0a8 2b01 a7b1 0035 0037 f800 064f 0100 ..+.5.7.0.0.
    0x0030:  0001 0000 0000 0000 1263 0fe6 0e65 0374 .....connect
    0x0040:  6976 6974 792d 0368 6563 b606 7562 756e ivity-check.ubun
    0x0050:  7475 0363 0fd0 0000 0100 01 tu.com.....
01:24:00.588551 IP localhost.34230 > localhost.domain: 37519+ [1au] PTR? 1.43.1
68.192.in-addr.arpa. (54)
    0x0000:  0000 0304 0006 0000 0000 0000 0000 0800 .....'.
    0x0010:  4500 0052 5bb9 4000 4011 e0ab 7f00 0001 E..R..@.0.0.0.
    0x0020:  7f00 0035 85b6 0035 003e fe85 928f 0120 ...5...5.2.0.0.0.
    0x0030:  0001 0000 0000 0001 0131 0234 3303 3136 .....1.43.16
    0x0040:  3803 3139 3207 696e 2d01 6464 7204 6172 8.192.in-addr.ar
    0x0050:  7061 0000 0c00 0100 0029 04b0 0000 0000 pa.....).
    0x0060:  0000 .
01:24:00.589057 IP anila-VirtualBox.38702 > 192.168.43.1.domain: 49972+ PTR? 1.
43.168.192.in-addr.arpa. (43)
    0x0000:  0004 0001 0006 0800 2705 7bdc 0000 0800 .....'{.....
    0x0010:  4500 0047 888e 4000 4011 ba5f 0a00 020f E..G..@.0.0.0.
    0x0020:  c0a8 2b01 972e 0035 0033 f7fc c334 0100 ..+.5.3..4..
```

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

Step1: \$sudo apt-get install netcat

```
anila@anila-VirtualBox:~$ sudo apt-get install netcat
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libdouble-conversion3
libgstreamer-plugins-bad1.0-0 libminizip1 libpcres2-16-0 libqt5core5a
libqt5dbus libqt5gui5 libqt5multimedia5 libqt5multimedialagstools5
libqt5multimediaWidgets5 libqt5network5 libqt5opengl5 libqt5svg5
libqt5widgets5 libva-wayland2 libxcb-xinerama0 libxcb-xinput0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
netcat
0 upgraded, 1 newly installed, 0 to remove and 314 not upgraded.
Need to get 2,172 B of archives.
After this operation, 15.4 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 netcat all 1.206-1ubuntu1 [2,172 B]
Fetched 2,172 B in 0s (3,094 B/s)
Selecting previously unselected package netcat.
(Reading database ... 191409 files and directories currently installed.)
Preparing to unpack .../netcat_1.206-1ubuntu1_all.deb ...
Unpacking netcat (1.206-1ubuntu1) ...
Setting up netcat (1.206-1ubuntu1) ...
```

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

```
anila@anila-VirtualBox:~$ nc -h
OpenBSD netcat (Debian patchlevel 1.206-1ubuntu1)
usage: nc [-46CDfhklNnrStUvVz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w
timeouts]
          [-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
```

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

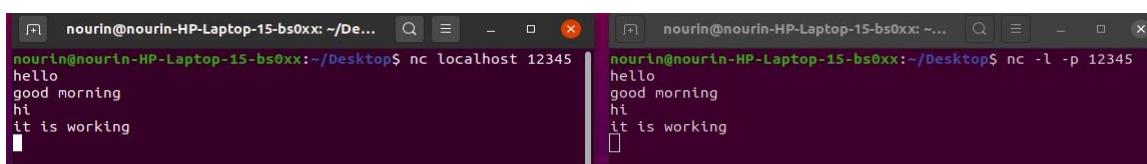
We will use port 12345 and will specify the port number with -p option.

```
anila@anila-VirtualBox:~$ nc -l -p 12345
hello
hai
```

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'

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



The image shows two terminal windows side-by-side. Both windows have a dark background with light-colored text. The left window shows the command 'nc localhost 12345' being run, followed by the message 'it is working'. The right window shows the command 'nc -l -p 12345' being run, followed by the same message 'it is working'. The terminals are identical in appearance, with standard Linux-style icons at the top.

```
nourin@nourin-HP-Laptop-15-bs0xx:~/Desktop$ nc localhost 12345
it is working
nourin@nourin-HP-Laptop-15-bs0xx:~/Desktop$ nc -l -p 12345
it is working
```

EXPERIMENT- 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
7735	02:24:31.867294	8.8.8.8	172.28.31.33	TLSv1.3	85	Application Data
7736	02:24:31.867374	172.28.31.33	8.8.8.8	TCP	54	63517 > 443 [ACK] Seq=2712 Ack=11670 Win=66048 Len=0
7737	02:24:31.867576	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.168? Tell 172.20.30.158
7738	02:24:31.867577	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.169? Tell 172.20.30.158
7739	02:24:31.867827	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.170? Tell 172.20.30.158
7740	02:24:31.868063	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.171? Tell 172.20.30.158
7741	02:24:31.868313	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.172? Tell 172.20.30.158
7742	02:24:31.868557	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.173? Tell 172.20.30.158
7743	02:24:31.868845	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.174? Tell 172.20.30.158
7744	02:24:31.869046	Hewlett_P_2e:67:07	Broadcast	ARP	68	Who has 172.20.55.175? Tell 172.20.30.158
7745	02:24:31.877027	8.8.4.4	172.28.31.33	DNS	233	Standard query response 0xb23d A urlite.ff.avast.com CNAME urlinfo.ns1.ff.avast.com A 69.94.67.99 A 5.62.40.110 A 5...
7746	02:24:31.877726	172.28.31.33	69.94.67.99	TCP	66	49520 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7747	02:24:31.880580	52.84.3.2	172.28.31.33	TCP	66	443 > 49519 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1280 SACK_PERM=1 WS=512
7748	02:24:31.880719	172.28.31.33	52.84.3.2	TCP	54	49519 > 443 [ACK] Seq=1 Ack=1 Win=66560 Len=0
7749	02:24:31.882561	8.8.8.8	172.28.31.33	TCP	68	443 > 63517 [ACK] Seq=11678 Ack=2712 Win=75264 Len=0
7750	02:24:31.883520	172.28.31.33	52.84.3.2	TLSv1.3	571	Client Hello
7751	02:24:31.883894	52.84.3.2	172.28.31.33	TCP	68	443 > 49519 [ACK] Seq=1 Ack=518 Win=66560 Len=0
7752	02:24:31.886674	8.8.4.4	172.28.31.33	DNS	233	Standard query response 0xd8d3 A urlite.ff.avast.com CNAME urlinfo.ns1.ff.avast.com A 69.94.67.99 A 5.62.40.110 A 5...
7753	02:24:31.887241	172.28.31.33	69.94.67.99	TCP	66	49521 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7754	02:24:31.890556	8.8.4.4	172.28.31.33	DNS	233	Standard query response 0x99ed A urlite.ff.avast.com CNAME urlinfo.ns1.ff.avast.com A 69.94.67.99 A 5.62.40.110 A 5...
7755	02:24:31.891120	172.28.31.33	69.94.67.99	TCP	66	49522 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7756	02:24:31.897277	172.28.31.33	65.8.79.226	TCP	66	59713 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7757	02:24:31.903983	172.28.31.33	8.8.8.8	TLSv1.3	243	Application Data
7758	02:24:31.908137	52.84.3.2	172.28.31.33	TLSv1.3	1314	Server Hello, Change Cipher Spec, Application Data

- 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 **02:09:02.092367**

No.	Time	Source	Destination	Protocol	Length	Info
611	02:09:02.092367	172.28.31.33	128.119.245.12	HTTP	661	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
635	02:09:02.392557	128.119.245.12	172.28.31.33	HTTP	293	HTTP/1.1 304 Not Modified
Frame 635: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits) on interface 0						
> Interface id: 0 {Device\NPF_{C369F20-6027-4FBF-87ED-1A214EE51ABD}}						
Encapsulation type: Ethernet (1)						
Arrival Time: Sep 23, 2021 02:09:02.392557000 Pacific Daylight Time						
[Time shift for this packet: 0.000000000 seconds]						
Epoch Time: 1632388142.392557000 seconds						
[Time delta from previous captured frame: 0.008239000 seconds]						
[Time delta from previous displayed frame: 0.300190000 seconds]						
[Time since reference or first frame: 5.520354000 seconds]						
Frame Number: 635						
Frame Length: 293 bytes (2344 bits)						
Capture Length: 293 bytes (2344 bits)						

The same section for the HTTP OK shows an arrival time of **02:09:02.392557**

The difference of these 2 times gives :

$$0.392557 - 0.092367 = 0.300190$$

- 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 = **172.20.10.2**

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

No.	Time	Source	Destination	Protocol	Length	Info
611	02:09:02.092367	172.20.31.33	128.119.245.12	HTTP	661	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
635	02:09:02.392557	128.119.245.12	172.20.31.33	HTTP	293	HTTP/1.1 304 Not Modified

- 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: Screenshot of HTTP GET message

No.	Time	Source	Destination	Protocol	Length	Info
11	21:46:25.593507	172.20.10.2	129.227.216.14	HTTP	256	GET /windows/ad-push.php?

channel=cd5e1e15&version=7.2.2.0 HTTP/1.1
Frame 11: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits) on interface \Device\NPF_{D990D16C-2693-4C28-8FE7-BBE2748DE72C}, id 0
Ethernet II, Src: IntelCor_5d:a7:e6 (d4:d2:52:5d:a7:e6), Dst: d6:dc:cd:8a:6b:64 (d6:dc:cd:8a:6b:64)
Internet Protocol Version 4, Src: 172.20.10.2, Dst: 129.227.216.14
Transmission Control Protocol, Src Port: 57354, Dst Port: 80, Seq: 1, Ack: 1, Len: 202
Hypertext Transfer Protocol

Screenshot of HTTP OK message+

No.	Time	Source	Destination	Protocol	Length	Info
13	21:46:26.115609	129.227.216.14	172.20.10.2	HTTP	446	HTTP/1.1 200 OK

Frame 13: 446 bytes on wire (3568 bits), 446 bytes captured (3568 bits) on interface \Device\NPF_{D990D16C-2693-4C28-8FE7-BBE2748DE72C}, id 0
Ethernet II, Src: d6:dc:cd:8a:6b:64 (d6:dc:cd:8a:6b:64), Dst: IntelCor_5d:a7:e6 (d4:d2:52:5d:a7:e6)
Internet Protocol Version 4, Src: 129.227.216.14, Dst: 172.20.10.2
Transmission Control Protocol, Src Port: 80, Dst Port: 57354, Seq: 1, Ack: 203, Len: 392
Hypertext Transfer Protocol

- 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 only sent 1 HTTP GET request to the server. The Packet that contained the GET message was packet number 4730
- The packet that contains the status code and phrase which the server sent in response to the GET message was packet number 4733
- The code and phrase in the response was 304 Not Modified

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
+ 611	02:09:02.092367	172.20.31.33	128.119.245.12	HTTP	661	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
- 635	02:09:02.392557	128.119.245.12	172.20.31.33	HTTP	293	HTTP/1.1 304 Not Modified

> Frame 611: 661 bytes on wire (5288 bits), 661 bytes captured (5288 bits) on interface 0
> Interface id: 0 (\Device\NPF_{C3C69F20-6D27-4FBF-87ED-1A214EE51ABD})
Encapsulation type: Ethernet (1)
Arrival Time: Sep 23, 2021 02:09:02.092367000 Pacific Daylight Time
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1632388142.092367000 seconds
[Time delta from previous captured frame: 0.002092000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 5.220164000 seconds]
Frame Number: 611
Frame Length: 661 bytes (5288 bits)
Capture Length: 661 bytes (5288 bits)

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

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
+ 611	02:09:02.092367	172.20.31.33	128.119.245.12	HTTP	661	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
- 635	02:09:02.392557	128.119.245.12	172.20.31.33	HTTP	293	HTTP/1.1 304 Not Modified

> Frame 635: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits) on interface 0
> Ethernet II, Src: Sophos_6b:b3:da (00:1a:8c:6b:b3:da), Dst: Hewlett_P_2e:68:5d (f4:39:09:2e:68:5d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.20.31.33
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 56052, Seq: 1, Ack: 608, Len: 239
Source Port: 80
Destination Port: 56052
[Stream index: 8]
[TCP Segment Len: 239]
Sequence number: 1 (relative sequence number)
[Next sequence number: 240 (relative sequence number)]
Acknowledgment number: 608 (relative ack number)
0101 = Header Length: 20 bytes (5)
> Flags: 0x010 (PSH, ACK)
Window size value: 238
[Calculated window size: 30464]
[Window size scaling factor: 128]
Checksum: 0x3b50 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
> [SEQ/ACK analysis]
> [Timestamps]
TCP payload (239 bytes)

CYCLE- 6

EXPERIMENT- 24

FAMILIARISATION TO HYPERVISORS AND VIRTUAL MACHINES

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 Benefits of hypervisor

- Speed
- Efficiency ➤ Flexibility ➤ Portability

Popular Hypervisors

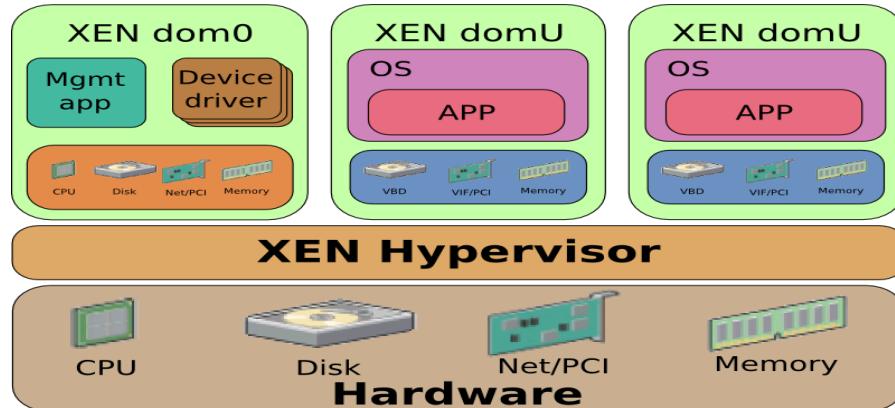
- Xen
- KVM(Kernel Based VM)
- SAN

EXPERIMENT-25

FAMILIARISATION TO XEN OR KVM

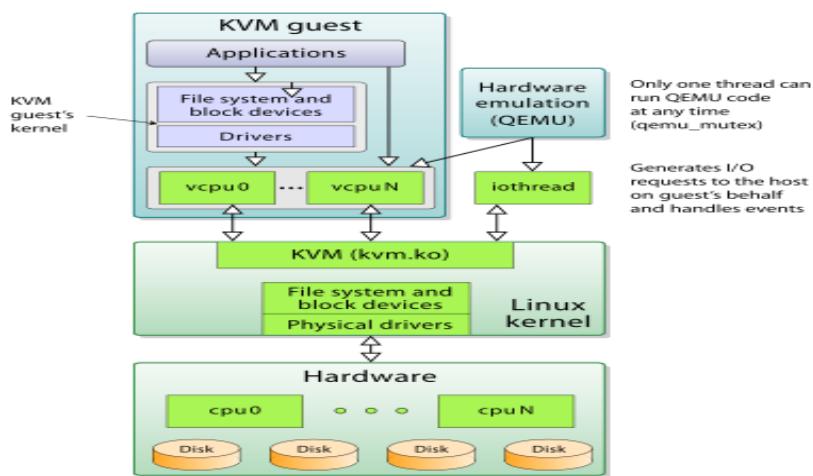
Xen

Xen is an open-source **paravirtualization** technology that provides a platform for running multiple operating systems in parallel on one physical hardware resource.



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.



DOCKER

- 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.

Docker installation

- \$ sudo apt install docker.io

```
anila@anila-VirtualBox:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libdouble-conversion3
  libgstreamer-plugins-bad1.0-0 libminizip1 libpcre2-16-0 libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimediasetools5
  libqt5multimediawidgets5 libqt5networks libqt5opengl5 libqt5svg5
  libqt5widgetss5 libvba-wayland2 libxcb-xinerama0 libxcb-xinput0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 314 not upgraded.
Need to get 74.0 MB of archives.
After this operation, 359 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1
[57.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1
.6-2ubuntu1 [30.5 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1
.0-rc95.ubuntu1.20.04.2 [4.087 kB]
```

- Version check**

```
$ docker --version
```

```
anila@anila-VirtualBox:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
anila@anila-VirtualBox:~$
```

- Check whether it is running or not**

```
$ sudo systemctl status docker
```

```

anila@anila-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese>
   Active: active (running) since Fri 2021-09-24 01:37:57 IST; 2min 39s ago
     TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
   Main PID: 4054 (dockerd)
      Tasks: 9
     Memory: 51.1M
        CGroup: /system.slice/docker.service
                  └─4054 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
Sep 24 01:37:52 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:52.4471>
Sep 24 01:37:52 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:52.4481>
Sep 24 01:37:52 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:52.4481>
Sep 24 01:37:52 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:52.4486>
Sep 24 01:37:54 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:54.1290>
Sep 24 01:37:54 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:54.5749>
Sep 24 01:37:56 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:56.9872>
Sep 24 01:37:56 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:56.9882>
Sep 24 01:37:57 anila-VirtualBox systemd[1]: Started Docker Application Contai>
Sep 24 01:37:57 anila-VirtualBox dockerd[4054]: time="2021-09-24T01:37:57.3346>
lines 1-21/21 (END)

```

- If not active

\$ sudo systemctl enable –now docker

- List all the images you have locally

\$ sudo docker images

```

anila@anila-VirtualBox:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
anila@anila-VirtualBox:~$
```

- Pull an image from the Docker registry

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

- Run a docker

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

```

anila@anila-VirtualBox:~$ sudo docker run python:latest
Unable to find image 'python:latest' locally
latest: Pulling from library/python
955615a668ce: Pull complete
2756ef5f69a5: 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:e6654afa815122b13242fc9ff513e2d14b00548ba6eaf4d3b03f2f261d85272d
Status: Downloaded newer image for python:latest
anila@anila-VirtualBox:~$
```

- List all the running containers

\$ sudo docker ps -a

EXPERIMENT- 26

INSTALLING 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” define 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

```
anila@anila-VirtualBox:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [101 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [538 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [538 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,222 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [874 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [283 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.3 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [634 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [855 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [354 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [18.8 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:16 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [10.4 kB]
```

Step 2: Enter

sudo apt install build-essential

```
anila@anila-VirtualBox:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libdouble-conversion3
libgstreamer-plugins-bad1.0-0 libminizip1 libpcre2-16-0 libqt5core5a
libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimediasupport5
libqt5multimediawidgets5 libqt5network5 libqt5opengl5 libqt5svg5
libqt5widget5 libv4l2 libxcb-xinerama0 libxcb-xinput0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++
g++-9 gcc gcc-10-base gcc-9 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasans libatomic1
libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
libctf0 libfakeroot libgcc-9-dev libgcc-s1 libgomp1 libitm1 liblsan0
libquadmath0 libstdc++-9-dev libstdc++6 libtsan0 libubsan1 linux-libc-dev
make manpages-dev
Suggested packages:
binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc
gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-9-multilib
gcc-9-locales glibc-doc libstdc++-9-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++-9 gcc gcc-9 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasans libatomic1
```

Step 3: Enter cd

/usr/local/src

/

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

Step 4: Enter sudo wget <http://www.noip.com/client/linux/noip-duc-linux.tar.gz>

```
anila@anila-VirtualBox:/usr/local/src$ sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
[sudo] password for anila:
--2021-09-25 11:35:13-- 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 23.6KB/s in 5.6s

2021-09-25 11:35:21 (23.6 KB/s) - 'noip-duc-linux.tar.gz' saved [134188/134188]

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

Step 5: Enter sudo tar xf noip-duc-linux.tar.gz and cd

noip-2.1.9-1/

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

Step 6: Enter sudo

make install

EXPERIMENT- 27

DEPLOY LINUX VM USING ANSIBLE PLAYBOOK

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

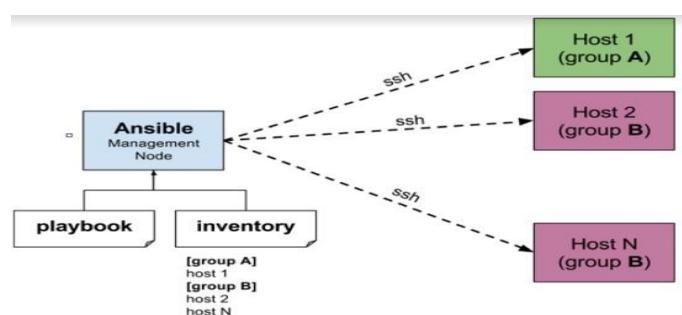
i.e. YAML (It's a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored)which is very easy for humans to understand, read and write. Hence the advantage is that even the IT infrastructure support guys can 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
```

```
anila@anila-VirtualBox:~/Desktop$ sudo apt-add-repository ppa:ansible/ansible
[sudo] password for anila:
Ansible is a radically simple IT automation platform that makes your applications and systems 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
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18.0 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:6 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,221 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [538 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [282 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,855 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27.6 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages
```

```
sudo apt update
```

```
anila@anila-VirtualBox:~/Desktop$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:6 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
310 packages can be upgraded. Run 'apt list --upgradable' to see them.
anila@anila-VirtualBox:~/Desktop$
```

```
sudo apt install ansible
```

```

anila@anila-VirtualBox:~/Desktop$ sudo apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libdouble-conversion3
  libgstreamer-plugins-bad1.0-0 libminizip1 libpcre2-16-0 libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimediasupports5
  libqt5multimediawidgets5 libqt5networks5 libqt5opengl5 libqt5svg5
  libqt5widgets5 libva-wayland2 libxcb-xinerama0 libxcb-xinput0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-distutils
  python3-dnspython python3-jinja2 python3-jmespath python3-kerberos
  python3-libzto3 python3-libcloud python3-netaddr python3-ntlm-auth
  python3-requests-kerberos python3-requests-ntlm python3-selinux
  python3-wiinrm python3-xmltodict
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-jinja2 python3-jmespath python3-kerberos
  python3-libcloud python3-netaddr python3-ntlm-auth
  python3-requests-kerberos python3-requests-ntlm python3-selinux

```

Setting up Inventory File 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?

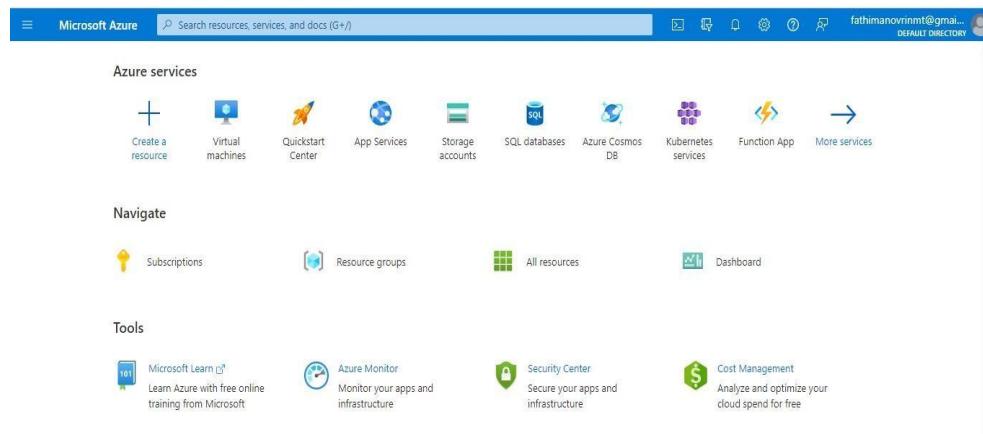
- Microsoft Azure consists of numerous service offerings, its use cases are extremely diverse. Running virtual machines or containers in the cloud is one of the most popular uses for Microsoft Azure. These compute resources can host infrastructure components, such as domain name system (DNS) servers; 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.

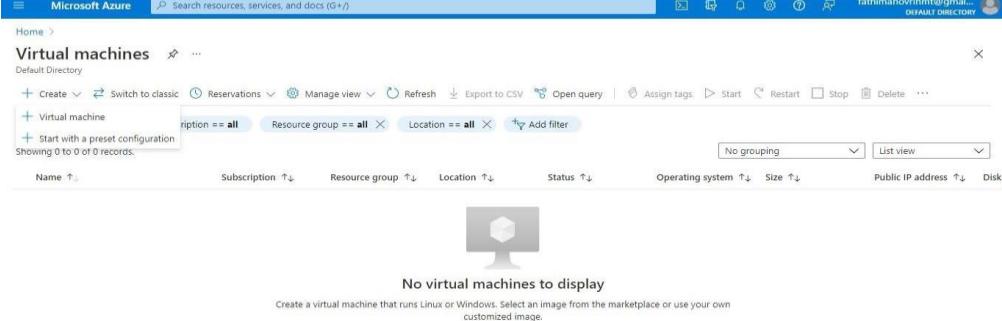
HOME PAGE OF AZURE



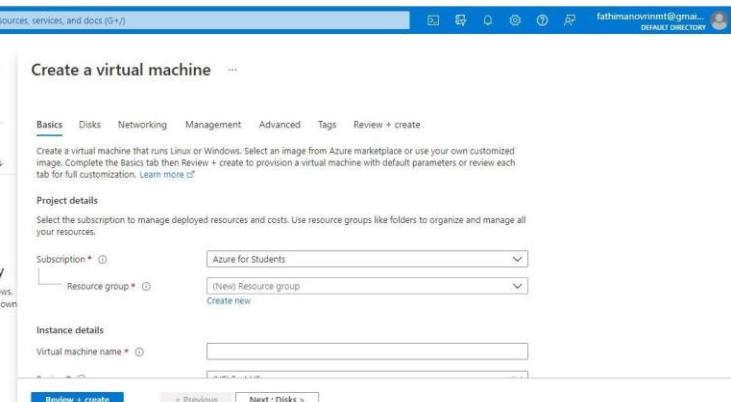
VM DEPLOYMENT WITHOUT ANSIBLE

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

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



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. Below it, the 'Virtual machines' section is displayed. A search bar at the top of this section shows filters: 'Subscription == all', 'Resource group == all', and 'Location == all'. Below the search bar, there are sorting options: 'Name ↑', 'Subscription ↑', 'Resource group ↑', 'Location ↑', 'Status ↑', 'Operating system ↑', 'Size ↑', 'Public IP address ↑', and 'Disks'. A large message in the center says 'No virtual machines to display' with a sub-instruction: 'Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.' At the bottom of the page, there are links for 'Learn more about Windows virtual machines' and 'Learn more about Linux virtual machines'.



The screenshot shows the 'Create a virtual machine' wizard on the 'Basics' tab. The left sidebar lists 'Virtual machines' and 'Create'. The main area has tabs for 'Basics', 'Disks', 'Networking', 'Management', 'Advanced', 'Tags', 'Review + create', and 'Create'. The 'Basics' tab contains fields for 'Subscription' (set to 'Azure for Students') and 'Resource group' (set to '(New) Resource group'). Below these are sections for 'Project details' (with a note about resource groups) and 'Instance details' (with a field for 'Virtual machine name'). At the bottom are buttons for 'Review + create', '< Previous', and 'Next : Disks >'.

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

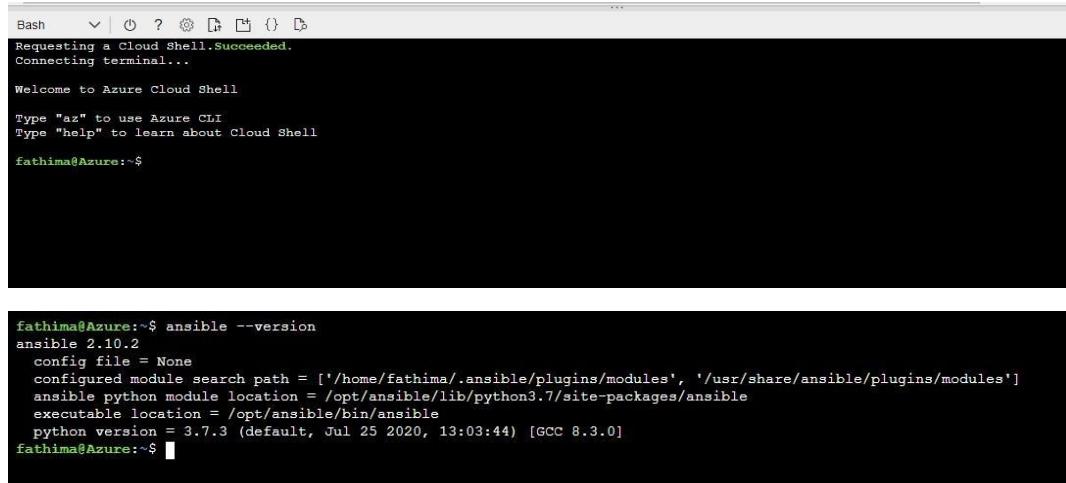
# Install Ansible. pip3 install "ansible==2.9.17"

# Install Ansible azure_rm module for interacting with Azure.

pip3 install ansible[azure]

```

To check the ansible version installed, **ansible --version**



```

Bash v | ⚡ ? 🌐 🔍 ⌂ ⌂ ⌂ ⌂ ⌂ ⌂
Requesting a Cloud Shell. Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell
fathima@Azure:~$ 

fathima@Azure:~$ ansible --version
ansible 2.10.2
  config file = None
  configured module search path = ['/home/fathima/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /opt/ansible/lib/python3.7/site-packages/ansible
  executable location = /opt/ansible/bin/ansible
  python version = 3.7.3 (default, Jul 25 2020, 13:03:44) [GCC 8.3.0]
fathima@Azure:~$ 

```

Creating a Linux vm(using ansible)

CREATING A RESOURCE GROUP

- name: Create Linux

VMhosts: localhost connection:

local

tasks:

- name: Create resource group

to hold

VMazure_rm_resourcegroup: name:

ansible-test-rg-vm location: eastus

CREATING A VIRTUAL NETWORK

```

- 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

```

CREATING A PUBLIC IP

```

-     name: Create public IP
  addressazure_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
  createddebug: msg: "The public IP is {{ output_ip_address.state.ip_address }}."

```

CREATING A NETWORK SECURITY GROUP

```

-     name: Create Network Security Group that allows
  SSHazure_rm_securitygroup:
  resource_group: ansible-test-rg-vm name:
    myNetworkSecurityGroup
  rules:
-     name: SSHprotocol: Tcp destination_port_range: 22
  access: Allow priority: 1001 direction: Inbound

```

CREATING A VIRTUAL NIC

```
- 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

```

CREATING A VM

```

- name: Create VM azure_rm_virtualmachine:

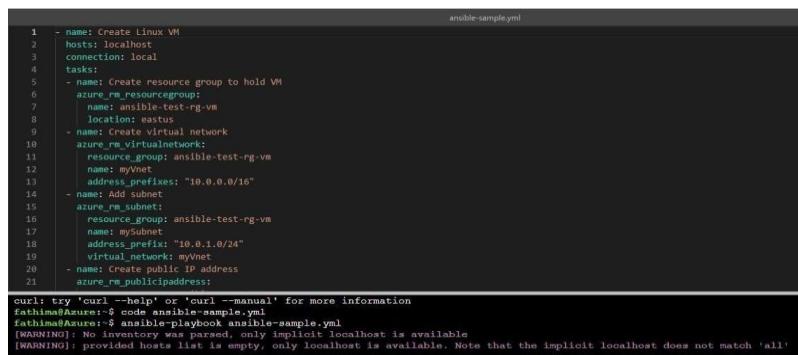
resource_group: ansible-test-rg-vm
name: Linuxvm vm_size:
Standard_DS1_v2 admin_username:
azureuser admin_password:
Pa55w.rd1234
ssh_password_enabled: true
network_interfaces: myNIC image:
offer: CentOS publisher:
OpenLogic

sku: '7.5'
version: latest

```

TO RUN THIS CODE,

- Open an azure shell and type :
- >**code ansible-filename.yml.**(It creates an ansible playbook where we enter the code above and save it)



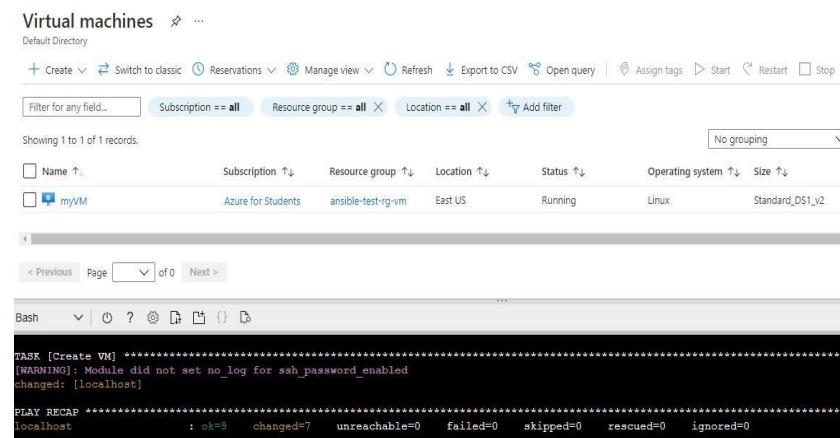
```

ansible@fathima-Azure:~$ code ansible-sample.yml
ansible@fathima-Azure:~$ ansible-playbook ansible-sample.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'.

```

- To run the above playbook created, use command:

->ansible-playbook ansible-filename.yml



The screenshot shows the Azure portal's 'Virtual machines' blade. It displays a single VM named 'myVM' with the following details:

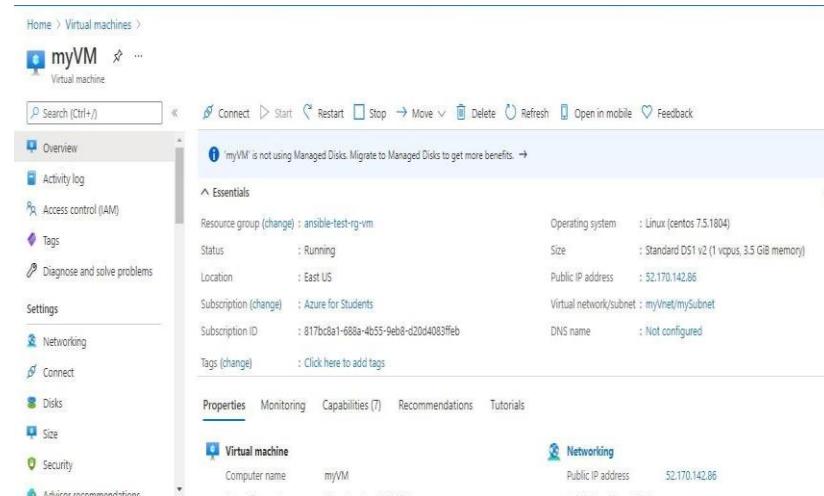
Name	Subscription	Resource group	Location	Status	Operating system
myVM	Azure for Students	ansible-test-rg-vm	East US	Running	Linux

Below the table, there is a terminal window titled 'Bash' showing the output of an Ansible playbook run:

```
TASK [Create VM] *****
[WARNING]: Module did not set no_log for ssh_password_enabled
changed: [localhost]

PLAY RECAP *****
localhost      : ok=9   changed=7   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```

THE VM CREATED WILL BE LIKE:

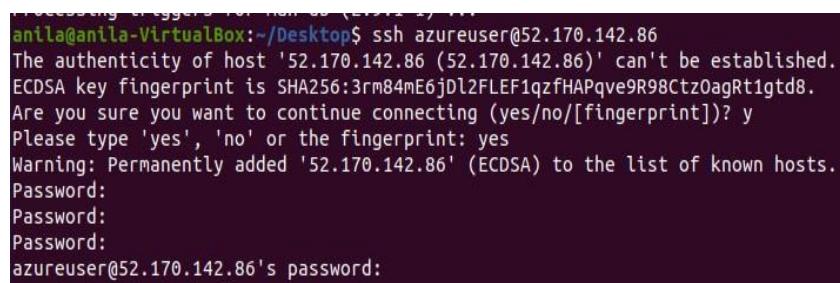


The screenshot shows the Azure portal's 'Virtual machine' details page for 'myVM'. The 'Overview' tab is selected. Key information displayed includes:

- Resource group: ansible-test-rg-vm
- Status: Running
- Location: East US
- Subscription: Azure for Students
- Subscription ID: 817bc8a1-688a-4b55-9eb8-d20d4083ffeb
- Tags: Click here to add tags
- Properties: Computer name: myVM
- Networking: Public IP address: 52.170.142.86

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

username@ public ip address of vm



```
anila@anila-VirtualBox:~/Desktop$ ssh azureuser@52.170.142.86
The authenticity of host '52.170.142.86 (52.170.142.86)' can't be established.
ECDSA key fingerprint is SHA256:3rm84mE6jDl2FLEF1qzfHAPqe9R98CtzOagRt1gtd8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '52.170.142.86' (ECDSA) to the list of known hosts.
Password:
Password:
Password:
azureuser@52.170.142.86's password:
```