

2oMCA136

**NETWORKING AND SYSTEM
ADMINISTRATION
LAB RECORD**

Vidhya Prasad

RMCA-S2-B

RollNo: 38

MOTHERBOARD COMPONENTS

1. Components of a Motherboard:

The motherboard is the main system board for the computer and connects all of the internal hardware components. This lesson will take a look at various components which are built into the motherboard. This lesson will also look at the expansion slots used to add hardware components to a system. The lesson finishes up with a discussion of the BIOS. The motherboard determines the capabilities and limitations of a computer system. Every component on a computer system plugs into the motherboard is controlled by it and depends on it to communicate. The motherboard houses the following components:

- **Central Processing Unit (CPU)** – performs all basic arithmetic, logical, control, and input/output operations
- **Chipset** – manages the data flow between the computer's processor, its memory and any peripheral devices attached
- **Buses** – a pathway that transfers data between components within a computer
- **Random Access Memory (RAM)** – a temporary form of computer data storage that allows fast access to data. RAM is volatile and requires power to keep data accessible.
- **Expansion slots** – provide expansion capability to add hardware components beyond what was originally installed
- **Ports** – provides an interface between the computer and a peripheral device such as a mouse, keyboard, or printer

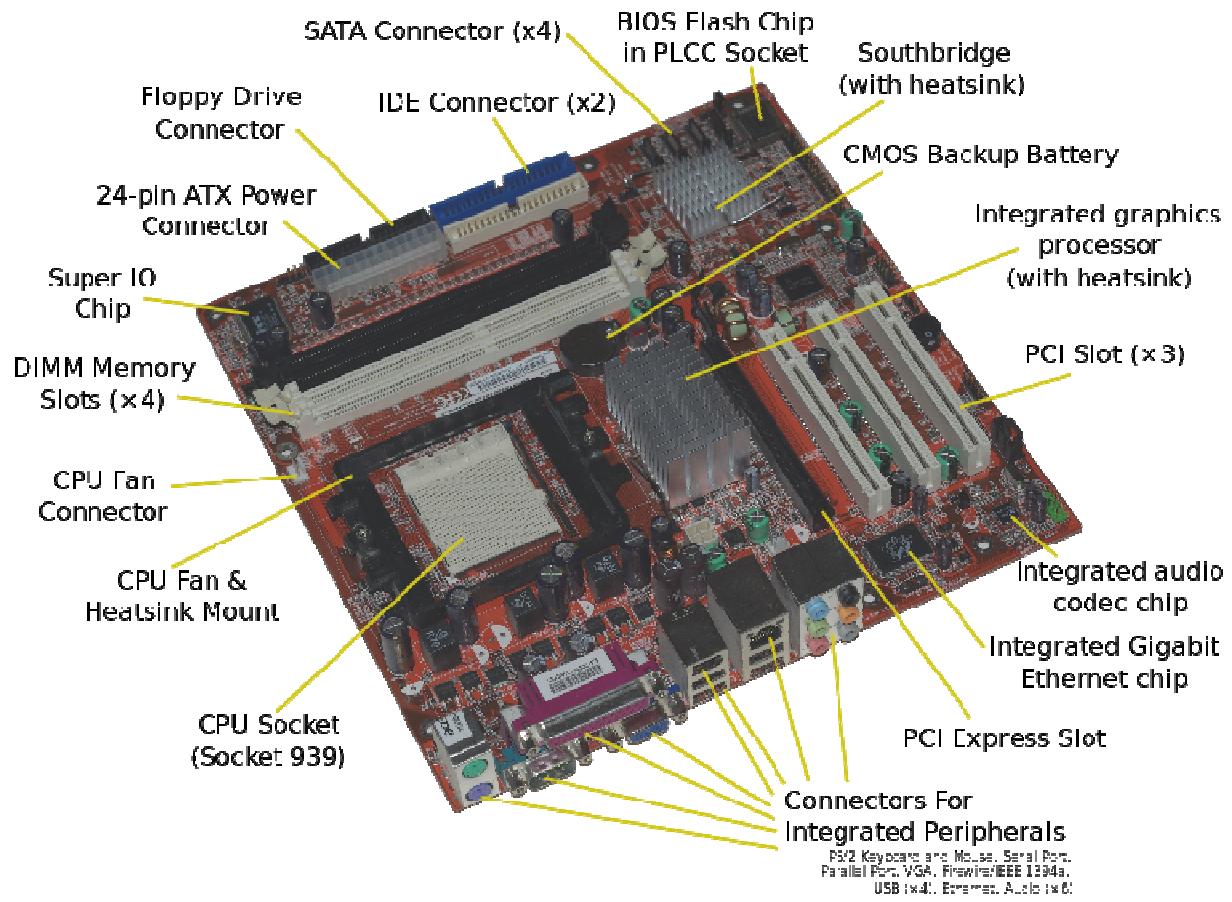


Figure 1: Motherboard Diagram with all components labeled

Expansion Slots

ISA slots. These were the oldest expansion slots in the history of motherboards. They were found in AT boards and are identified by black color. Conventional [display cards](#) or sound cards were installed in these slots. The full form of ISA is **Industry Standard Architecture** and is a 16-bit bus.

PCI Slots. The full form of PCI is Peripheral Component Interconnect. The PCI slot is one of the important motherboard components today and is vastly used to install [add-on cards](#) on the motherboard. The PCI supports 64-bit high-speed bus.

PCI express. Also known as PCIe, these are the latest and the fastest component of the motherboard to support add-on cards. It supports full duplex serial bus.

AGP slot. Accelerated graphics port (AGP) is specifically used to install a latest graphics card. AGP runs on a 32-bit bus and both PCIe and AGP can be used to install high-end gaming display cards

RAM(memory) slots

SIMM slots. The full form is a single in-line [memory](#) module. These slots were found in older motherboards, up to 486-boards. The SIMM supports 32-bit bus.

DIMM slots. The full form of DIMM is a Double inline memory module. These are the latest [RAM](#) slots which run on a faster 64-bit bus. The DIMM used on Laptop boards are called SO-DIMM.

CPU Socket

Another vital motherboard component is the [CPU](#) socket which is used to install the processor on the motherboard. Some important sockets are explained below.

Socket7. It is a 321 pin socket that supported older processors like Intel Pentium 1/2/MMX, AMD k5/K6, and Cyrix M2.

Socket370. It is a 370 pin socket that supports Celeron processors and Pentium-3 processors.

Socket 775. It is a 775-pin socket that supports Inter dual core, C2D, P-4 and Xeon processors.

Socket 1156. Found on latest types of motherboards, it is an 1156-pin socket that supports latest Intel i-3, i-5 and i-7 processors.

Socket 1366. The socket is of 1366 pins and supports latest i-7 900 processors.

BIOS

The full form of BIOS is Basic Input Output System. It is a motherboard component in the form of a Integrated chip. This chip contains all the information and settings of the motherboard which you can modify by entering the BIOS mode from your computer.

CMOS Battery

The battery or a cell is a 3.0 Volts lithium type cell. The cell is responsible for storing the information in BIOS and the full form is Complementary Metal Oxide Semi-Conductor.

Power Connectors

In order to receive power from [SMPS](#), there are connectors mounted on the motherboards.

AT connector. It consists of 2 number of 6 pin male connectors and is found on old types of motherboards.

ATX connector. The latest in the series of power connectors, they are either 20 or 24 pin female connectors. Found in all the latest types of motherboards.

IDE connector

The Integrated Drive Electronics (IDE) connectors are used to interface disk drives. The 40-pin male connector is used to connect IDE [hard disk drives](#) and the 34-pin male connector connects to Floppy Disk Drive.

SATA connector

Latest in the series, the connectors, Serial Advance Technology Attachment(SATA) are 7-pin connectors to interface latest SATA hard disks or [optical drives](#). They are much faster than IDE interface.

Co-Processor

The co-processor is one of the important motherboard components and helps the main processor in mathematical calculations and computer graphics.

Cabinet connections

The cabinet in which the motherboard is installed has many buttons that connect to the motherboard. Some of the common connectors are Power Switch, Reset Switch, Front [USB](#), Front Audio, Power indicator(LED) and HDD LED.

2. Ram Modules:

Virtually all memory modules use some type of dynamic RAM (**DRAM**) chips. DRAM requires frequent recharges of memory to retain its contents.

SRAM

Static random-access memory (SRAM) is RAM that does not need to be periodically refreshed. Memory refreshing is common to other types of RAM and is basically the act of reading information from a specific area of memory and immediately rewriting that information back to the same area without modifying it. Due to SRAM's architecture, it does not require this refresh. You will find SRAM being used as cache memory for CPUs, as buffers within hard drives, and as temporary storage for LCD screens. Normally, SRAM is soldered directly to a printed circuit board (PCB) or integrated directly to a chip. This means that you probably won't be replacing SRAM. SRAM is faster than—and is usually found in smaller quantities than—its distant cousin DRAM.

SDRAM

Synchronous DRAM (SDRAM) was the first type of memory to run in sync with the processor bus (the connection between the processor, or CPU, and other components on the motherboard). Most 168-pin DIMM modules use SDRAM memory. To determine whether a DIMM module contains SDRAM memory, check its speed markings. SDRAM memory is rated by bus speed (PC66 equals 66MHz bus speed; PC100 equals 100MHz bus speed; and PC133 equals 133MHz bus speed). All SDRAM modules have a one-bit prefetch buffer and perform one transfer per clock cycle.

Depending on the specific module and motherboard chipset combination, PC133 modules can sometimes be used on systems that are designed for PC100 modules.

DDR SDRAM

The second generation of systems running synchronous DRAM use double data rate SDRAM (DDR SDRAM). **DDR SDRAM** performs two transfers per clock cycle (instead

of one, as with regular SDRAM) and features a two-bit prefetch buffer. 184-pin DIMM memory modules use DDR SDRAM chips.

While DDR SDRAM is sometimes rated inMHz, it is more often rated by throughput (MBps). Common speeds for DDR SDRAM include PC1600 (200MHz/1600Mbps), PC2100 (266MHz/2100Mbps), PC2700 (333MHz/2700Mbps), and PC3200 (400MHz/3200Mbps), but other speeds are available from some vendors.

DDR2 SDRAM

Double data rate 2 SDRAM (DDR2 SDRAM) is the successor to DDR SDRAM. DDR2 SDRAM runs its external data bus at twice the speed of DDR SDRAM and features a four-bit prefetch buffer, enabling faster performance. However, DDR2 SDRAM memory has greater latency than DDR SDRAM memory. Latency is a measure of how long it takes to receive information from memory; the higher the number, the greater the latency. Typical latency values for mainstream DDR2 memory are CL=5 and CL=6, compared to CL=2.5 and CL=3 for DDR memory. 240-pin memory modules use DDR2 SDRAM.

DDR2 SDRAM memory might be referred to by the effective memory speed of the memory chips on the module (the memory clock speed x4 or the I/O bus clock speed x2)—for example, DDR2-533 (133MHz memory clock x4 or 266MHz I/O bus clock x2)=533MHz—or by module throughput (DDR2-533 is used in PC2-4200 modules, which have a throughput of more than 4200Mbps). PC2- indicates the module uses DDR2 memory; PC- indicates the module uses DDR memory.

Other common speeds for DDR2 SDRAM modules include PC2-3200 (DDR2-400; 3200Mbps throughput); PC2-5300 (DDR2-667); PC2-6400 (DDR2-800); and PC2-8500 (DDR2-1066).

DDR3 SDRAM

Double data rate 3 SDRAM (DDR3 SDRAM) Compared to DDR2, DDR3 runs at lower voltages, has twice the internal banks, and most versions run at faster speeds than DDR2. DDR3 also has an eight-bit prefetch bus. As with DDR2 versus DDR, DDR3 has

greater latency than DDR2. Typical latency values for mainstream DDR3 memory are CL7 or CL9, compared to CL5 or CL6 for DDR2. Although DDR3 modules also use 240 pins, their layout and keying are different than DDR2, and they cannot be interchanged.

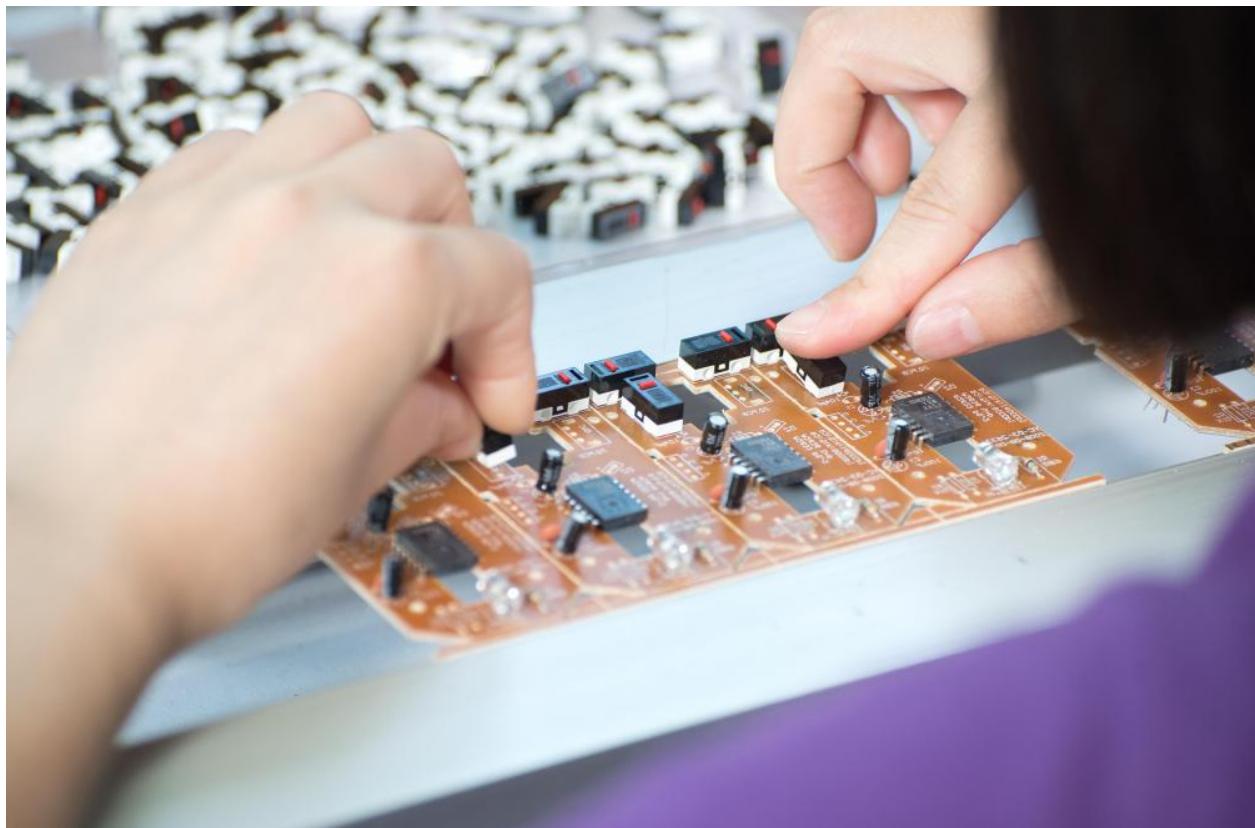
DDR3 SDRAM memory might be referred to by the effective memory speed of the memory chips on the module (the memory clock speed x4 or the I/O bus clock speed x2); for example, DDR3-1333 (333MHz memory clock x4 or 666MHz I/O bus clock x2)=1333MHz) or by module throughput (DDR3-1333 is used in PC3-10600 modules, which have a throughput of more than 10,600MBps or 10.6GBps). PC3- indicates the module uses DDR3 memory.

Other common speeds for DDR3 SDRAM modules include PC3-8500 (DDR3-1066; 8500MBps throughput); PC3-12800 (DDR3-1600); and PC3-17000 (DDR3-2133).

3. Daughter cards:

A daughter card or **daughterboard** is a type of **circuit board** that gets added to an existing one. Its name is appropriate for its use, since it is connected to a “**motherboard**” or “main board.” The motherboard is the primary circuit board for a device. It is usually in the device as it is shipped from the factory. A daughter card may be added later. Some daughter card designs are made so that engineers can add functionality to a device without requiring a lot more room inside its housing. These kinds of items are often called riser boards or risers. Some might also call them “mezzanine boards.” With the rise of connective USB ports and other technology, it has become less necessary to upgrade devices with daughter cards or daughter boards. A lot of advanced use can be built into a wireless connection and “outsourced” to a remote server, rather than adding it physically into a desktop or laptop computer. However, some types of equipment might still get these kinds of additions as provided by the manufacturer. Computer and electronics makers choose the best ways of offering upgrades that they feel will match the needs and desires of their customer base. Since

not a lot of laptop or computer users want to wrestle a daughter card into an existing circuit board design, companies that sell to a consumer market will probably choose alternatives, or offer professional installation as a free service if they are offering a daughter card as a way to upgrade a device.



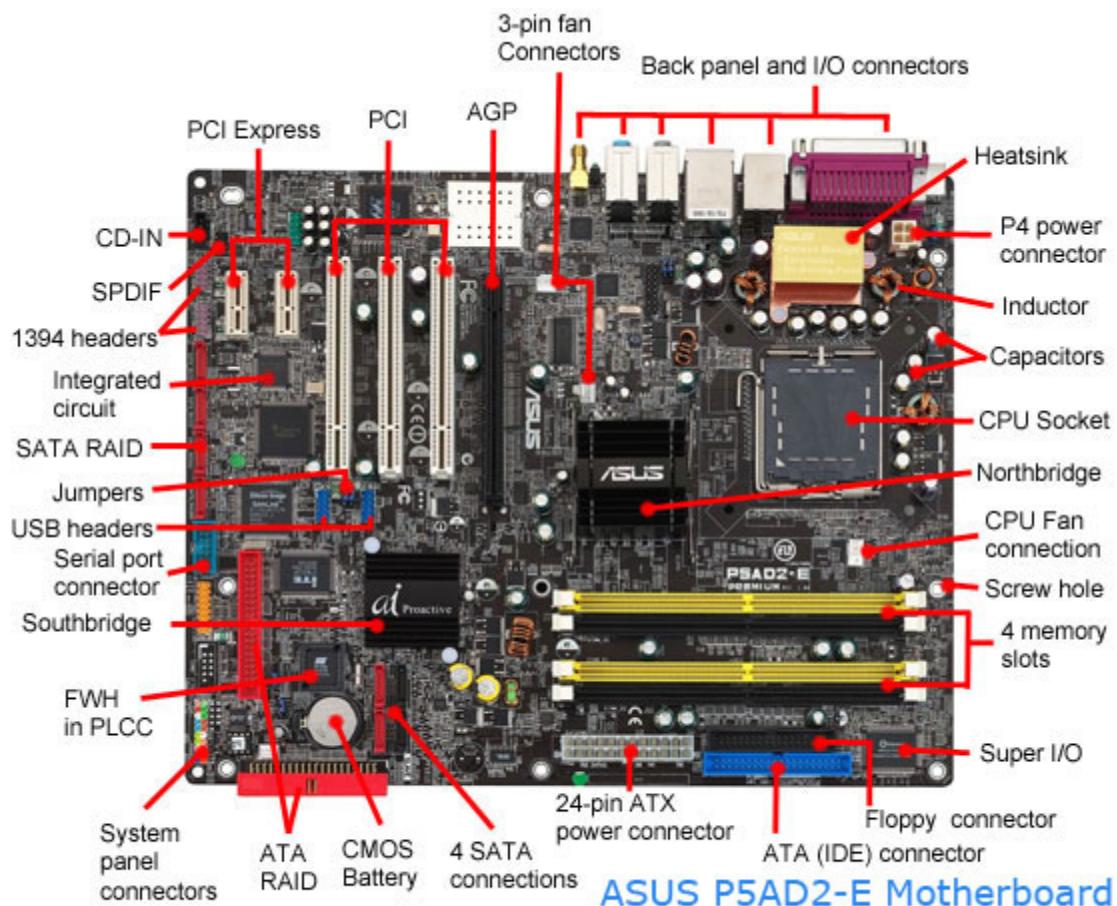
4. Bus Slots:

Alternatively known as a **bus slot** or **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.

Below is a listing of expansion slots commonly found in a computer and the devices associated with those slots. Clicking any of the links below provide you with additional details.

- [AGP - Video card.](#)
- [AMR - Modem, sound card.](#)
- [CNR - Modem, network card, sound card.](#)
- [EISA - SCSI](#), network card, video card.
- [ISA - Network card, sound card, video card.](#)
- [PCI - Network card, SCSI, sound card, video card.](#)
- [PCI Express - Video card, modem, sound card, network card.](#)
- [VESA - Video card.](#)

Many of the expansion card slots above are obsolete. You're most likely only going to encounter AGP, PCI, and PCI Express when working with computers today. The picture below is an example of what expansion slots may look like on a motherboard. In this picture, there are three different types of expansion slots: PCI Express, PCI, and AGP.



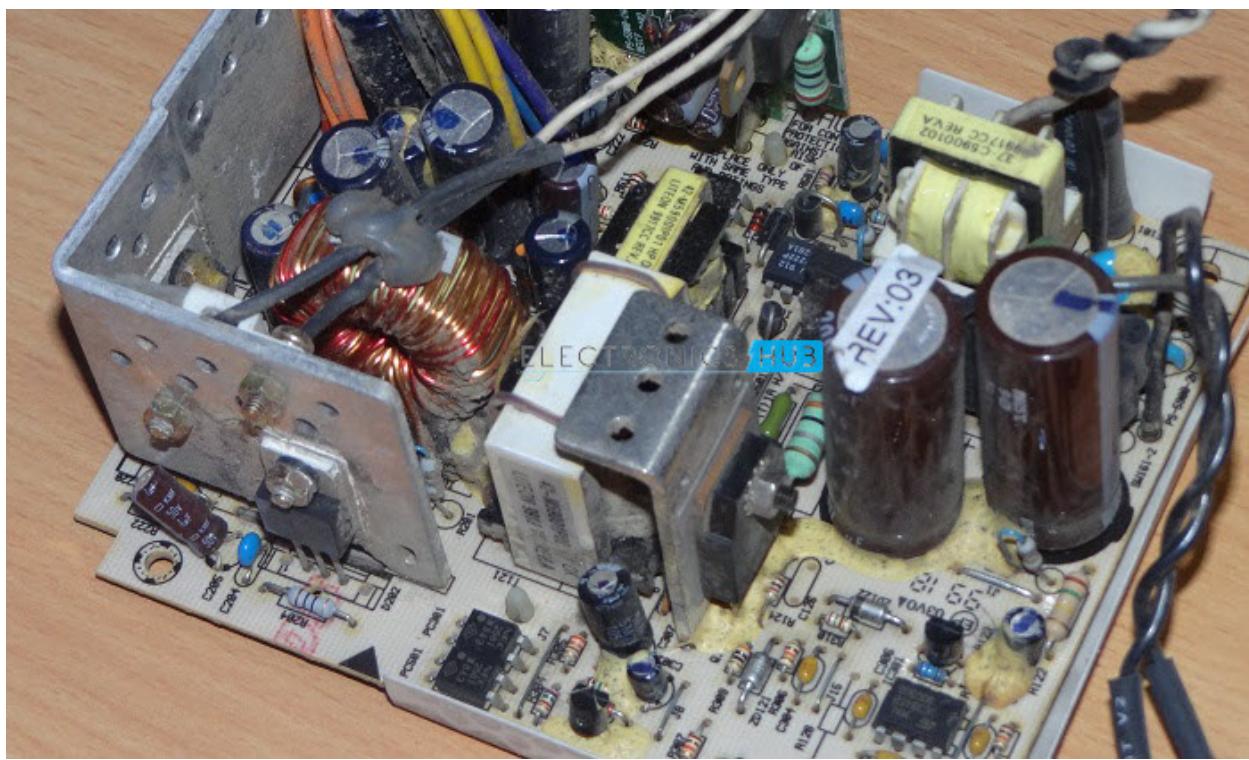
ASUS P5AD2-E Motherboard

ComputerHope.com

5. SMPS:

Switch Mode or Switching Mode Power Supply or simply SMPS is a type of Power Supply Unit (PSU) that uses some kind of switching devices to transfer electrical energy from source to load. Usually the source is either AC or DC and the load is DC. The most common application of an SMPS is the power supply unit of a computer. Switching Mode Power Supply (SMPS) has become a standard type of power supply unit for electronic devices because of their high efficiency, low cost and high power density.

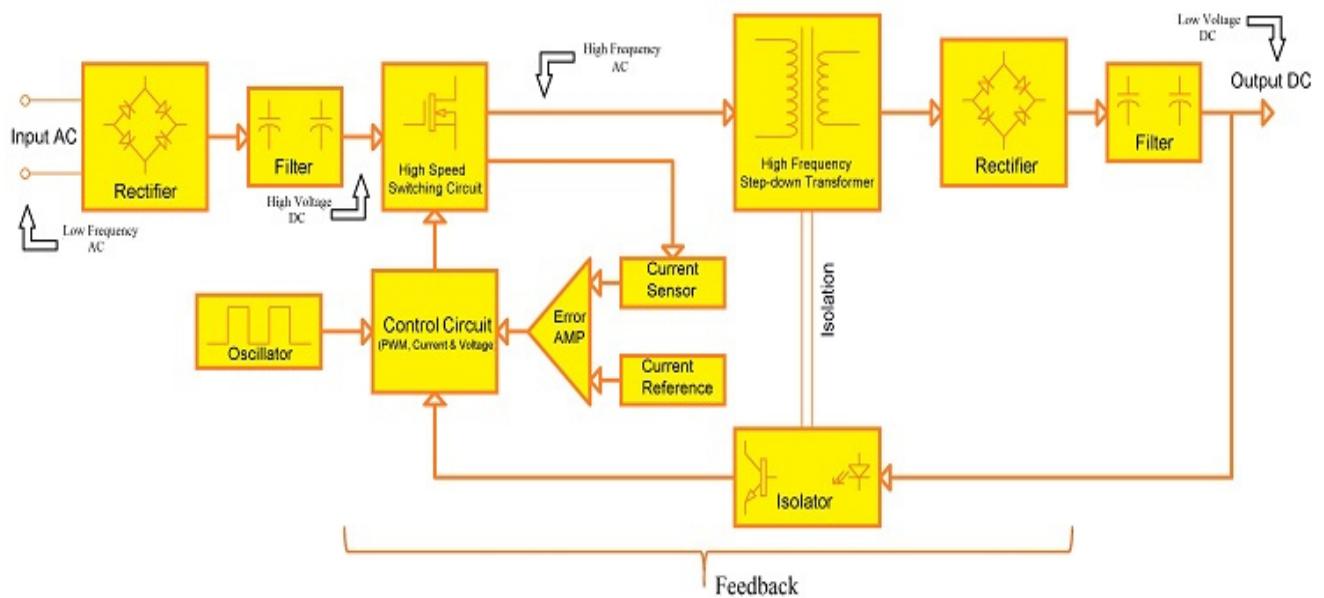
The following image shows an SMPS unit from an old desktop computer. This particular SMPS is rated for 90W of power.



Majority of Electronic DC Loads like Microprocessors, Microcontrollers, LEDs, Transistors, ICs, Motors etc. are supplied with standard power sources like batteries for

example. Unfortunately, the prime problem with batteries is the voltage is either too high or too low. Hence, an SMPS will provide a regulated DC output. SMPS is a versatile power supply as we can choose from different topologies like Step – up (Boost), Step – down (Buck), power supplies with isolation at input and output depending on the type of application. Coming to the major factor of why we need SMPS, the efficiency of a good SMPS design can be as high as 90% or even more. In contrast, the efficiency of a Linear Regulated Power Supply is dependent on the voltage drop at the pass transistor. For example, assume we have a 3V Lithium Cell that must be stepped down to a 1.8V load drawing a current of 100mA. The power wasted in the transistor as heat is 0.12W and hence the efficiency of the power supply is 40%. SMPS ICs come with more or less all the features of a discrete SMPS design allowing engineers to experiment with design for custom projects. The design of Switched Mode Power Supply or SMPS is fairly complex when compared to linear regulated power supply. But this complexity in design has an advantage as it will result in stable and regulated DC supply that is capable of delivering more power in an efficient way for a given physical specification (size, weight and cost).

A simplified block diagram of an SMPS which converts AC input to a regulated DC is shown in the following image.



Although there are many number of design types for an SMPS power supply, all the designs will be more or less similar to the structure shown above. The main design types in SMPS are:

- AC to DC, where AC mains is given as input and we get a regulated DC at the output,
- DC to DC Step up converter, where an input DC voltage is stepped up i.e. output voltage is greater than input and
- DC to DC Step down converter, where the input DC voltage is stepped down i.e. output voltage is less than or equal to input voltage.

In case of DC to DC SMPS systems, the input DC is usually given from a battery and hence, both the DC to DC converter circuits (Step up and Step down) are commonly found in battery operated systems. Coming back to SMPS design in the above image, it represents a typical AC to DC converter. We will see the basic working of this SMPS design. The input AC supply is given to rectifier and filter circuits. This step will convert the High Voltage AC to High Voltage DC. This high voltage DC is given to a High Speed Switching Element like a Power MOSFET. The output of this switch, which is a High Frequency, High Voltage Pulsating AC, is given to a High Frequency Step down Transformer. The output of this transformer is a Low Voltage AC signal which is in turn given to a rectifier and a filter circuit to obtain Low Voltage DC.

6. Internal Storage Devices:

Internal storage devices, also called "main or primary memory" refers to memory that stores small amounts of data that can be accessed quickly while the computer is running. There are basically two kinds of internal memory: ROM and RAM. **ROM** stands for read-only memory. It is non-volatile, which means it can retain data even without power. It is used mainly to start or boot up a computer.

Once the operating system is loaded, the computer uses **RAM**, which stands for random-access memory, which temporarily stores data while the central processing unit (CPU) is executing other tasks. With more RAM on the computer, the less the CPU has to read data from the external or secondary memory (storage device), allowing the computer to run faster. RAM is fast but it is volatile, which means it will not retain data if there is no power. It is therefore important to save data to the storage device before the system is turned off.

Types of RAM

There are two main types of RAM: Dynamic RAM (DRAM) and Static RAM (SRAM).

- **DRAM** (pronounced DEE-RAM), is widely used as a computer's main memory. Each DRAM memory cell is made up of a transistor and a capacitor within an integrated circuit, and a data bit is stored in the capacitor. Since transistors always leak a small amount, the capacitors will slowly discharge, causing information stored in it to drain; hence, DRAM has to be refreshed (given a new electronic charge) every few milliseconds to retain data.
- **SRAM** (pronounced ES-RAM) is made up of four to six transistors. It keeps data in the memory as long as power is supplied to the system unlike DRAM, which has to be refreshed periodically. As such, SRAM is faster but also more expensive, making DRAM the more prevalent memory in computer systems.

Types of DRAM

- **Synchronous DRAM (SDRAM)** “synchronizes” the memory speed with CPU clock speed so that the memory controller knows the exact clock cycle when the requested data will be ready. This allows the CPU to perform more instructions at a given time. Typical SDRAM transfers data at speeds up to 133 MHz.
- **Rambus DRAM (RDRAM)** takes its name after the company that made it, Rambus. It was popular in the early 2000s and was mainly used for video game devices and graphics cards, with transfer speeds up to 1 GHz.

- **Double Data Rate SDRAM (DDR SDRAM)** is a type of synchronous memory that nearly doubles the bandwidth of a single data rate (SDR) SDRAM running at the same clock frequency by employing a method called "double pumping," which allows transfer of data on both the rising and falling edges of the clock signal without any increase in clock frequency.
- **DDR1 SDRAM** has been succeeded by **DDR2, DDR3**, and most recently, **DDR4 SDRAM**. Although operating on the same principles, the modules are not backward-compatible. Each generation delivers higher transfer rates and faster performance. The latest DDR4 modules, for example, feature fast transfer rates at 2133/2400/2666 and even 3200 MT/s.

7. Interfacing Ports:

A port is a physical docking point using which an external device can be connected to the computer. It can also be programmatic docking point through which information flows from a program to the computer or over the Internet.

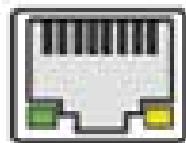
A port has the following characteristics –

- External devices are connected to a computer using cables and ports.
- Ports are slots on the motherboard into which a cable of external device is plugged in.
- Examples of external devices attached via ports are the mouse, keyboard, monitor, microphone, speakers, etc.

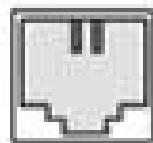
IES Power Connectors



Ethernet / RJ45



Modem / RJ11



eSata

External Hard Drive Port



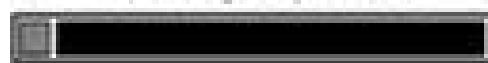
DisplayPort

Video and Audio Port for Home Theater Systems



PCMCIA / Cardbus

Modem, Networking and Expansion Cards



VGA Port

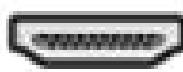
Pinouts and icons



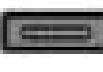
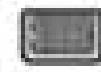
S - Video



HDMI



Digital Video Interface



Serial Port



Parallel Port



PS/2 Port



Games Port



Legend: Audio Mini - Jacks Symbols

▲ Microphone

● Stereo Line-In

◆ Stereo Line-Out

○ Right to Left

— Center / Subwoofer



Serial Port

- Used for external modems and older computer mouse
- Two versions: 9 pin, 25 pin model
- Data travels at 115 kilobits per second

Parallel Port

- Used for scanners and printers
- Also called printer port
- 25 pin model
- IEEE 1284-compliant Centronics port

PS/2 Port

- Used for old computer keyboard and mouse
- Also called mouse port
- Most of the old computers provide two PS/2 port, each for the mouse and keyboard
- IEEE 1284-compliant Centronics port

Universal Serial Bus (or USB) Port

- It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.
- It was introduced in 1997.
- Most of the computers provide two USB ports as minimum.
- Data travels at 12 megabits per seconds.
- USB compliant devices can get power from a USB port.

VGA Port

- Connects monitor to a computer's video card.
- It has 15 holes.
- Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

Power Connector

- Three-pronged plug.
- Connects to the computer's power cable that plugs into a power bar or wall socket.

Firewire Port

- Transfers large amount of data at very fast speed.
- Connects camcorders and video equipment to the computer.
- Data travels at 400 to 800 megabits per seconds.
- Invented by Apple.
- It has three variants: 4-Pin FireWire 400 connector, 6-Pin FireWire 400 connector, and 9-Pin FireWire 800 connector.

Modem Port

- Connects a PC's modem to the telephone network.

Ethernet Port

- Connects to a network and high speed Internet.
- Connects the network cable to a computer.
- This port resides on an Ethernet Card.
- Data travels at 10 megabits to 1000 megabits per seconds depending upon the network bandwidth.

Game Port

- Connect a joystick to a PC
- Now replaced by USB

Digital Video Interface, DVI port

- Connects Flat panel LCD monitor to the computer's high-end video graphic cards.
- Very popular among video card manufacturers.

Sockets

- Sockets connect the microphone and speakers to the sound card of the computer.

Topic:

Basic Linux commands with example.

Contents:

1. pwd
2. history
3. man
4. ls
5. cd
6. mkdir
7. rmdir
8. touch
9. rm
10. cat

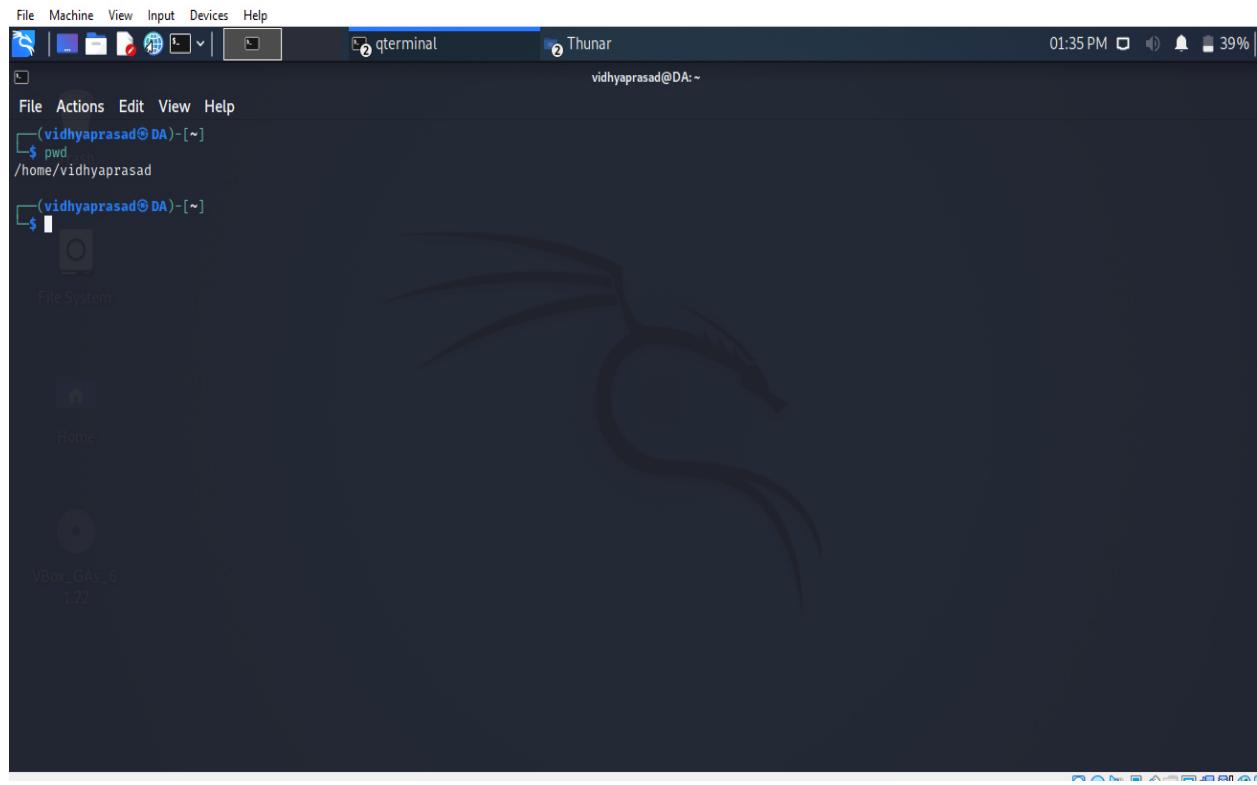
1. pwd:

Linux pwd (print working directory) command displays your location currently you are working on. It will give the whole path starting from the root ending to the directory.

Syntax:

```
$ pwd
```

Example:



2. history

Linux history command is used to display the history of the commands executed by the user. It is a handy tool for auditing the executed commands along with their date and time.

Most of the commands read input from the terminal a line at a time. But, the history command is capable of keeping the record of those lines with associated data. By default, it will show the last five hundred commands from the older to most recent commands. The history library is saved in a history file.

Syntax:

The basic syntax for the history command is as follows:

```
$ history
```

Example:

```
(vidhyaprasad@DA) [~]
$ history
 1  ls
 2  sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade
 3  sudo apt-get install linux-headers-$(uname -r)
 4  sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade
 5  sudo apt-get install linux-headers-$(uname -r)
 6  ls
 7  pwd
 8  history
 9  ls
10 man
11 clear
12 pwd
13 ls
14 man pwd
15 man
16 man pwd
17 ls
18 cd nwlab
19 clear
20 pwd
21 ls
22 man pwd
23 pwd
24 man pwd
25 ls
26 mkdir nwlab
27 ls
28 ls -r
29 ls -t
30 cd nwlab/
31 mkdir work1
32 ls
33 mkdir -p/ linuxcmd/cmd1
```

Display the nth command from the history

We can display the specific number of commands by specifying it as "`!<command number>`". For example, we want to show the most recent command which is 500th in our history file, execute the command as follows:

Syntax:

`! Command number`

Example:

```
89 history
90 touch file1 file2 file3
91 ls
92 touch file1 file2 file3
93 cd nwlab/
94 ls
95 cd nwlab/
96 touch file1 file2 file3
97 ls
98 man history
99 man ls
100 man man
101 man clear
102 history
103 history rm mycommands

[(vidhyaprasad@DA)-[~]
$ history !99

[(vidhyaprasad@DA)-[~]
$ history man ls]
```

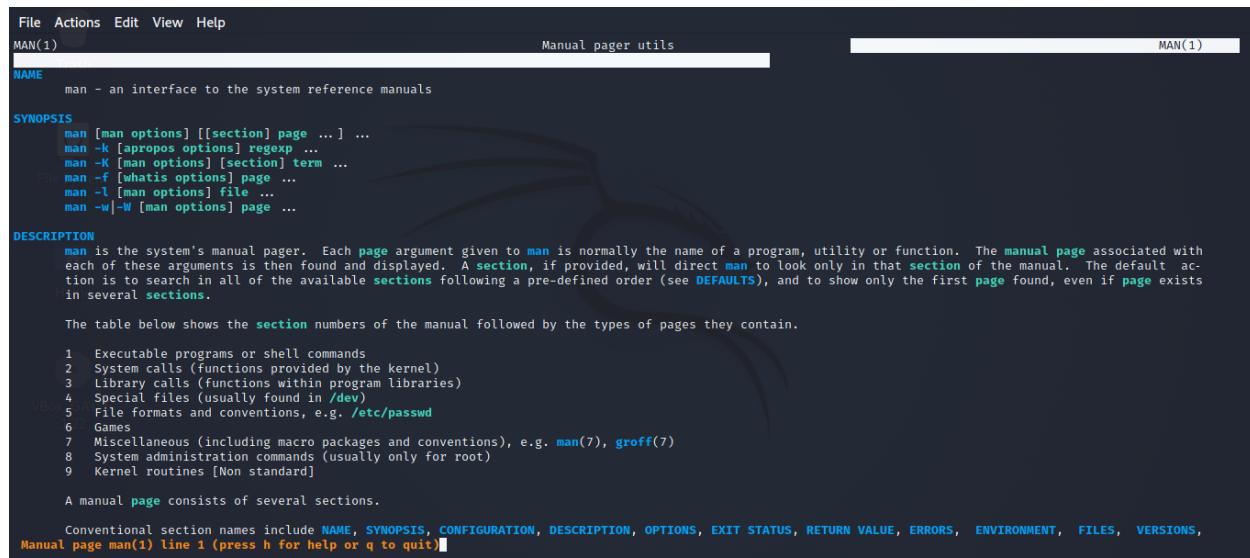
3. man:

man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

Syntax :

```
$man COMMAND NAME
```

Example:



The screenshot shows a terminal window with the title bar "MAN(1)" and the application name "Manual pager utils". The window displays the man(1) manual page. The content includes:

- NAME**: man - an interface to the system reference manuals
- SYNOPSIS**: man [man options] [[section] page ...] ...
man -k [apropos options] regexp ...
man -K [man options] [section] term ...
man -f [whatis options] page ...
man -l [man options] file ...
man -w|-W [man options] page ...
- DESCRIPTION**: man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.
- A table showing section numbers and their types:

Section	Type
1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in /dev)
5	File formats and conventions, e.g. /etc/passwd
6	Games
7	Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8	System administration commands (usually only for root)
9	Kernel routines [Non standard]

- A note about pages: A manual page consists of several sections.
- Conventional section names include NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUE, ERRORS, ENVIRONMENT, FILES, VERSIONS.
- Bottom status bar: Manual page man(1) line 1 (press h for help or q to quit).

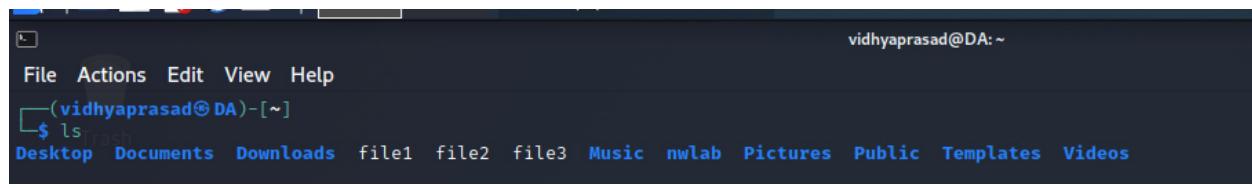
4. ls:

- **ls :** The **ls** is the list command in Linux. It will show the full list or content of your directory. Just type **ls** and press the enter key. The whole content will be shown.

Syntax:

```
$ ls
```

Example:



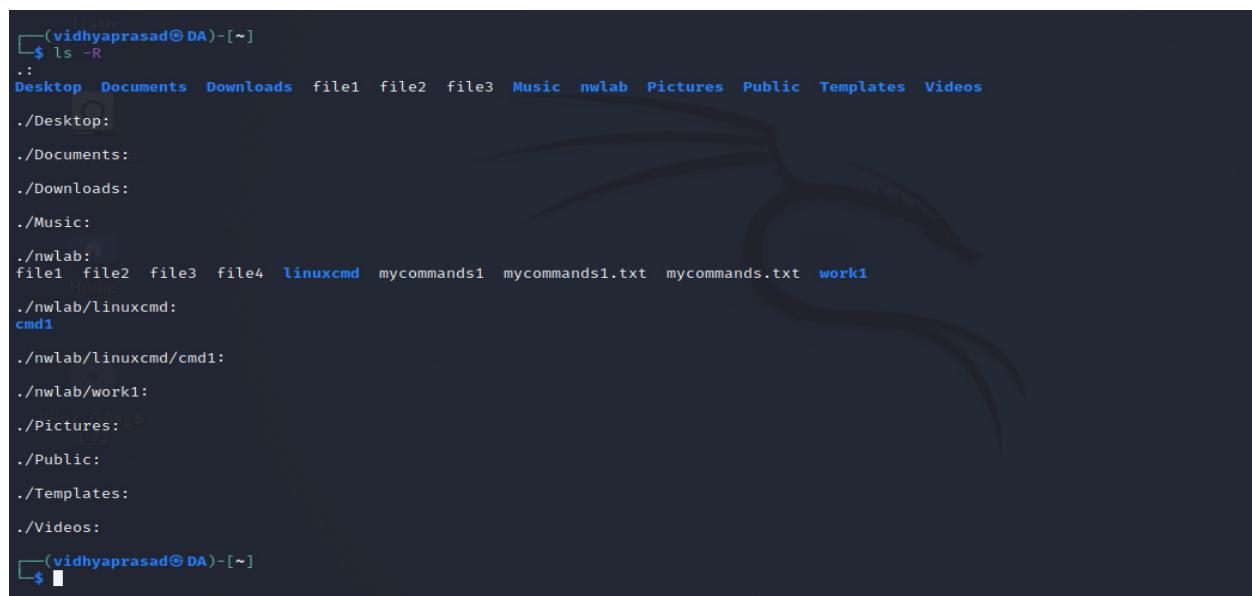
A screenshot of a terminal window titled 'Terminal'. The window has a dark background with light-colored text. At the top, there's a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu, it shows the user's name 'vidhyaprasad@DA' and the current directory '~'. The command '\$ ls' is typed at the prompt, followed by a list of files and directories: Desktop, Documents, Downloads, file1, file2, file3, Music, nwlab, Pictures, Public, Templates, and Videos.

- **ls -R :** option will list very long listing directory trees.

Syntax:

```
$ ls -R
```

Example:



A screenshot of a terminal window titled 'Terminal'. The window has a dark background with light-colored text. The command '\$ ls -R' is typed at the prompt. The output shows a recursive listing of all files and subdirectories from the current directory down to the deepest levels. This includes entries like '.', '..', and many sub-directories named after the user's files and commands.

- **ls -l** : Here, **ls -l** (-l is character not one) shows file or directory, size, modified date and time, file or folder name and owner of file and its permission.

Syntax:

```
$ ls -l
```

Example:

```
(vidhyaprasad@DA) [~]
$ ls -l
total 36
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 11:36 Desktop
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Documents
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Downloads
-rw-r--r-- 1 vidhyaprasad vidhyaprasad 0 Jun 12 18:56 file1
-rw-r--r-- 1 vidhyaprasad vidhyaprasad 0 Jun 12 18:56 file2
-rw-r--r-- 1 vidhyaprasad vidhyaprasad 0 Jun 12 18:56 file3
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Music
drwxr-xr-x 4 vidhyaprasad vidhyaprasad 4096 Jun 12 15:59 nwlab
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Pictures
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Public
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Templates
drwxr-xr-x 2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Videos

(vidhyaprasad@DA) [~]
$ █

VBox_GAs_6
1.22
```

- **ls -a** : List all files including hidden.

Syntax:

```
$ ls -a
```

Example:

```
(vidhyaprasad@DA) [~]
$ ls -a
. .bashrc.original .dmrc .face.icon .gnupg nwlab Templates .vboxclient-seamless.pid .xsession-errors.old
.. .cache Documents file1 .ICEauthority Pictures .vboxclient-clipboard.pid Videos .zsh_history
.bash_logout .config Downloads file2 .local .profile .vboxclient-display-svga-x11.pid .Xauthority .zshrc
.bashrc Desktop .face file3 Music Public .vboxclient-draganddrop.pid .xsession-errors
$ █
```

- **ls -al** : will list the files and directories with detailed information like the permission, size, owner, etc.

Syntax:

```
$ ls -al
```

Example:

```
└─(vidhyaprasad㉿DA)─[~]
$ ls -al
total 148
drwxr-xr-x 15 vidhyaprasad vidhyaprasad 4096 Jun 13 08:49 .
drwxr-xr-x  3 root      root      4096 Jun 12 10:30 ..
-rw-r--r--  1 vidhyaprasad vidhyaprasad  220 Jun 12 10:30 .bash_logout
-rw-r--r--  1 vidhyaprasad vidhyaprasad 5349 Jun 12 10:30 .bashrc
-rw-r--r--  1 vidhyaprasad vidhyaprasad 3526 Jun 12 10:30 .bashrc.original
drwxr-xr-x  6 vidhyaprasad vidhyaprasad 4096 Jun 13 08:48 .cache
drwx----- 9 vidhyaprasad vidhyaprasad 4096 Jun 12 10:56 .config
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 11:36 Desktop
-rw-r--r--  1 vidhyaprasad vidhyaprasad   55 Jun 12 10:57 .dmrc
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Documents
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Downloads
-rw-r--r--  1 vidhyaprasad vidhyaprasad 11759 Jun 12 10:30 .face
lrwxrwxrwx  1 vidhyaprasad vidhyaprasad    5 Jun 12 10:30 .face.icon → .face
-rw-r--r--  1 vidhyaprasad vidhyaprasad   0 Jun 12 18:56 file1
-rw-r--r--  1 vidhyaprasad vidhyaprasad   0 Jun 12 18:56 file2
-rw-r--r--  1 vidhyaprasad vidhyaprasad   0 Jun 12 18:56 file3
drwx----- 3 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 .gnupg
-rw----- 1 vidhyaprasad vidhyaprasad   0 Jun 12 10:36 .ICEauthority
drwxr-xr-x  3 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 .local
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Music
drwxr-xr-x  4 vidhyaprasad vidhyaprasad 4096 Jun 12 15:59 nwlab
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Pictures
-rw-r--r--  1 vidhyaprasad vidhyaprasad   807 Jun 12 10:30 .profile
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Public
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Templates
-rw-r----- 1 vidhyaprasad vidhyaprasad    4 Jun 13 08:48 .vboxclient-clipboard.pid
-rw-r----- 1 vidhyaprasad vidhyaprasad    4 Jun 13 08:48 .vboxclient-display-svga-x11.pid
-rw-r----- 1 vidhyaprasad vidhyaprasad    4 Jun 13 08:48 .vboxclient-draganddrop.pid
-rw-r----- 1 vidhyaprasad vidhyaprasad    4 Jun 13 08:48 .vboxclient-seamless.pid
drwxr-xr-x  2 vidhyaprasad vidhyaprasad 4096 Jun 12 10:36 Videos
-rw----- 1 vidhyaprasad vidhyaprasad    47 Jun 13 08:48 .Xauthority
-rw----- 1 vidhyaprasad vidhyaprasad 7069 Jun 13 08:49 .xsession-errors
```

- **ls -t:** list files sorted in the order of last modified.

Syntax:

```
$ ls -t
```

Example:

```
└─(vidhyaprasad@DA)─[~]
$ ls -t
file2  file3  file1  nwlab  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
└─(vidhyaprasad@DA)─[~]
$ █
```

- **ls -r :** option display files and directories in reverse order.

Syntax:

```
$ ls -r
```

Example:

```
└─(vidhyaprasad@DA)─[~]
$ ls -r
Videos  Templates  Public  Pictures  nwlab  Music  file3  file2  file1  Downloads  Documents  Desktop
└─(vidhyaprasad@DA)─[~]
$ █
```

- **ls -tr :** this will reverse the time wise listing.

Syntax:

```
$ ls -tr
```

Example:

```
└─(vidhyaprasad@DA)─[~]
$ ls -tr
Videos  Templates  Public  Pictures  Music  Downloads  Documents  Desktop  nwlab  file1  file3  file2
└─(vidhyaprasad@DA)─[~]
$ █
```

5. cd:

The `cd` (“change directory”) command is used to change the current working directory in Linux and other Unix-like operating systems. It is one of the most basic and frequently used commands when working on the Linux terminal.

Syntax:

`$cd directory name`

Example:

```
(vidhyaprasad@DA)~]$ pwd  
/home/vidhyaprasad  
(vidhyaprasad@DA)~]$ cd nwlab/  
(vidhyaprasad@DA)~/nwlab]$
```

When specifying a directory to change to, you can use either absolute or relative path names. The absolute or full path starts from the system root `/`, and relative path starts from your current directory. By default, when you log into your Linux system, your current working directory is set to your home directory. Assuming that `Downloads` directory exists in your home directory, you can navigate to it by using the relative path to the directory:

Example:

```
$ cd Downloads
```

You can also navigate to the same directory by using its absolute path:

```
$ cd /home/username/Downloads
```

In short, if the path starts with a slash (/) it is the absolute path to the directory.

- **cd..** : to move one directory up

Syntax:

```
$ cd..
```

Example:

```
└─(vidhyaprasad@DA)-[~/nwlab]
  $ cd ..
  └─(vidhyaprasad@DA)-[~]
    $ ls
    Desktop  Documents  Downloads  file1  file2  file3  Music  nwlab  Pictures  Public  Templates  Videos
  └─(vidhyaprasad@DA)-[~]
    $ pwd
    /home/vidhyaprasad
  └─(vidhyaprasad@DA)-[~]
    $
```

- **cd -** :to move to the previous directory.

Syntax:

```
$ cd -
```

Example:

```
└──(vidhyaprasad@DA)-[~]
  └─$ pwd
  /home/vidhyaprasad
  └──(vidhyaprasad@DA)-[~]
    └─$ cd ~
    ~/nwlab
    └──(vidhyaprasad@DA)-[~/nwlab]
      └─$ pwd
      /home/vidhyaprasad/nwlab
      └──(vidhyaprasad@DA)-[~/nwlab]
        └─$ █
```

- **cd** : to go straight to the home folder.

Syntax:

```
$ cd
```

Example:

```
└──(vidhyaprasad@DA)-[~/nwlab]
  └─$ cd
  └──(vidhyaprasad@DA)-[~]
    └─$ pwd
    /home/vidhyaprasad
    └──(vidhyaprasad@DA)-[~]
      └─$ █
```

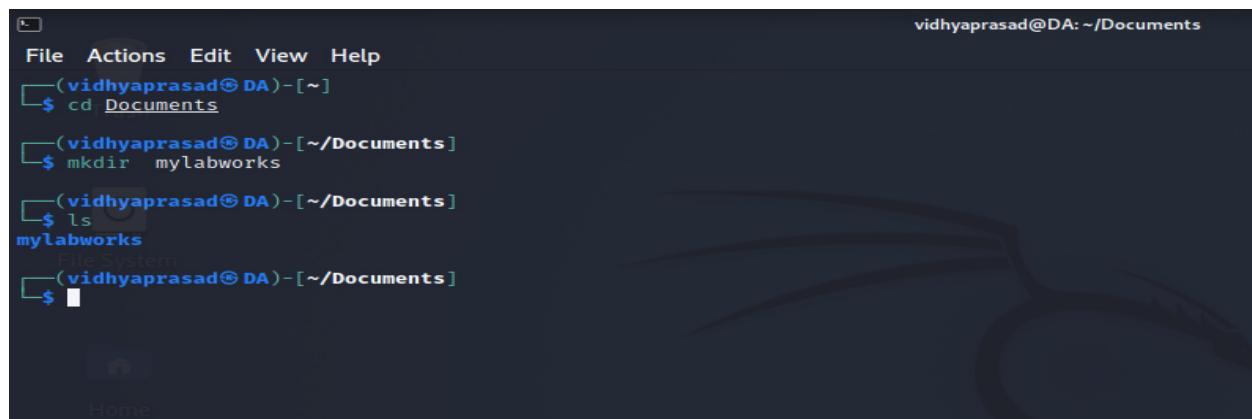
6. mkdir:

mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permissions to create a directory in the parent directory, or he/she may receive a ‘permission denied’ error.

Syntax:

```
$ mkdir directory name
```

Example:



```
File Actions Edit View Help
(vidhyaprasad@DA)-[~]
$ cd Documents
(vidhyaprasad@DA)-[~/Documents]
$ mkdir mylabworks
(vidhyaprasad@DA)-[~/Documents]
$ ls
mylabworks
$
```

- use the p option to create a directory in between two existing directories.

Syntax:

```
Mkdir -p diectoyname1/directoryname2
```

Example:

```
File System
└─(vidhyaprasad@DA)-[~/Documents]
  └─$ mkdir -p sample/sample1

└─(vidhyaprasad@DA)-[~/Documents]
  └─$ ls
  mylabworks  sample

└─(vidhyaprasad@DA)-[~/Documents]
  └─$ ls sample/
  sample1

└─(vidhyaprasad@DA)-[~/Documents]
  └─$ █
```

7. rmdir:

To remove a directory use the rmdir command, however it only allows you to delete empty directories.

Syntax:

Rmdir directory name/

Example:

```
Linux
└─(vidhyaprasad㉿DA)-[~/nwLab]
└─$ rmdir linuxcmd/
rmdir: failed to remove 'linuxcmd/': Directory not empty

└─(vidhyaprasad㉿DA)-[~/nwLab]
└─$ █
```

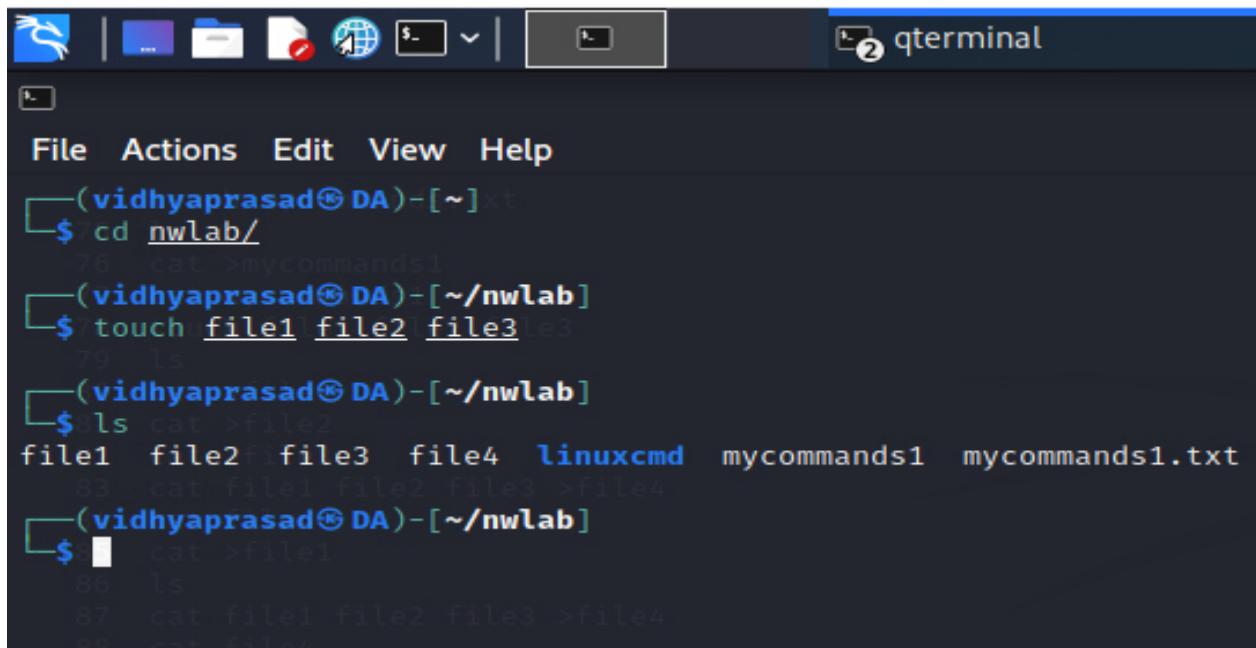
8. touch:

touch command is used to create a blank new file in linux.

Syntax:

Touch filename

Example:



The screenshot shows a Linux desktop environment with a terminal window titled "qterminal". The terminal window has a dark background and contains the following command-line session:

```
(vidhyaprasad@DA)-[~] xt
$ cd nwlab/
76 cat >mycommands1
(vidhyaprasad@DA)-[~/nwlab]
$ touch file1 file2 file3
79 ls
(vidhyaprasad@DA)-[~/nwlab]
$ ls cat >file2
file1 file2 file3 file4 linuxcmd mycommands1 mycommands1.txt
83 cat file1 file2 file3 >file4
(vidhyaprasad@DA)-[~/nwlab]
$ 84 cat >file1
86 ls
87 cat file1 file2 file3 >file4
88 cat file4
```

9. rm:

rm command is used to remove or delete directories and the contents within them.

Syntax:

rm filename

Example:

```
[~(vidhyaprasad@DA)-[~/nwlab]
$ ls
linuxcmd  mycommands.txt  mycommands.txt  work1
[~(vidhyaprasad@DA)-[~/nwlab]
$ rm mycommands.txt
[~(vidhyaprasad@DA)-[~/nwlab]
$ ls
mycommands.txt  work1
[~(vidhyaprasad@DA)-[~/nwlab]
$ █
```

10. cat:

The 'cat' command is the most universal and powerful tool. It is considered to be one of the most frequently used commands. It can be used to display the content of a file, copy content from one file to another, concatenate the contents of multiple files, display the line number, display \$ at the end of the line, etc.

Syntax:

cat file Name

Example:

```
└──(vidhyaprasad㉿DA)-[~/nwlab]
└$ touch mycommands.txt
Home
└──(vidhyaprasad㉿DA)-[~/nwlab]
└$ ls
linuxcmd  mycommands1.txt  mycommands.txt  work1

└──(vidhyaprasad㉿DA)-[~/nwlab]
└$ cat >mycommands1
hellow this is my lab work!
this will show u the basic linux commands!
^C .1.22

└──(vidhyaprasad㉿DA)-[~/nwlab]
└$ cat mycommands1
hellow this is my lab work!
this will show u the basic linux commands!

└──(vidhyaprasad㉿DA)-[~/nwlab]
└$ █
```

- cat commands also join two files and stores the result in another new file.

Syntax:

Cat filename1 filename2>filename3

Example:

```
vidhyaprasad@DA:~/Documents
File Actions Edit View Help
(vidhyaprasad@DA) [~]
$ cd Documents
(vidhyaprasad@DA) [~/Documents]
$ touch file1 file2 file3
(vidhyaprasad@DA) [~/Documents]
$ cat >file1
first file
^C File System
(vidhyaprasad@DA) [~/Documents]
$ cat >file2
second file
^C
(vidhyaprasad@DA) [~/Documents]
$ cat >file3
third file
^C
(vidhyaprasad@DA) [~/Documents]
$ cat file1 file2 file3 >file4
(vidhyaprasad@DA) [~/Documents]
$ cat file4
first file
second file
third file
(vidhyaprasad@DA) [~/Documents]
$
```

To Append the Content of A File

The 'cat' command with double greater than sign (**>>**) append (add something in the last of a file) something in your already existing file.

Syntax:

cat >> (file name)

- cat command is also used to convert a file to upper case or lower case

Syntax:

cat filename | tr a-z A-Z>output.txt

Example:

```
└─(vidhyaprasad@DA)-[~/Documents]
$ cat file4
first file
second file
third file

└─(vidhyaprasad@DA)-[~/Documents]
$ cat file4 | tr a-z A-Z >file5.txt

└─(vidhyaprasad@DA)-[~/Documents]
$ ls
file1  file2  file3  file4  file5.txt  mylabworks  sample

└─(vidhyaprasad@DA)-[~/Documents]
$ cat file5.txt
FIRST FILE
SECOND FILE
THIRD FILE

└─(vidhyaprasad@DA)-[~/Documents]
$ █
```

Topic:

Basic Linux commands with example.

Contents:

- 1. echo
- 2. head
- 3. tail
- 4. read
- 5. more
- 6. less
- 7. cut
- 8. paste
- 9. uname
- 10. copy
- 11. mv
- 12. locate
- 13. find
- 14. grep
- 15. df
- 16. du
- 17. useradd
- 18. userdel
- 19. sudo
- 20.passwd

1. echo:

echo command is used to move some data into a file.

Syntax:

```
$ echo data >>filename
```

Example:

```
[~(vidhyaprasad@DA)-[~/Documents]
$ echo my linux lab >>file1
[~(vidhyaprasad@DA)-[~/Documents]
$ cat file1
first file
my linux lab
[~(vidhyaprasad@DA)-[~/Documents]
$
```

2. 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.

Syntax:

```
$ head -n line number filename.txt
```

Example:

```
└─(vidhyaprasad@DA)-[~/Documents]
$ cat file4
first file
second file
third file
this is my second lab work
basic linux commands
this will show u the linux commands
my head command

└─(vidhyaprasad@DA)-[~/Documents]
$ head -n 4 file4
first file
second file
third file
this is my second lab work

└─(vidhyaprasad@DA)-[~/Documents]
```

3. 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.

Syntax:

```
$ tail -n filename.txt
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ cat file4
first file
second file
third file
this is my second lab work
basic linux commands
this will show u the linux commands
my head command
VBox_GA$_o

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ tail -n 4 file4
this is my second lab work
basic linux commands
this will show u the linux commands
my head command

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

4. read:

read the contents of a line into a variable. The read command can be used with and without arguments
read command is used to read [options] [name...]

Syntax:

```
$read var1 var2 var3
```

```
$echo "[${var1}] [${var2}] [${var3}]"
```

Example:

```
root@kali:~# read v1 v2 v3
my linux lab
root@kali:~# echo ["$v1"]["$v2"]["$v3"]
[my][linux][lab]
root@kali:~#
```

5. more:

Like cat command, more command displays the content of a file. Only difference is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.

Syntax:

```
$more /etc/passwd
```

Example:

```
root@kali:~# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
```

6. less:

The 'less' command is same as 'more' command but include some more features. • It automatically adjust with the width and height of the teminal window, while 'more' command cuts the content as the width of the terminal window get shorter.

Syntax:

```
$ less filename
```

Example:

Input

```
└─(vidhyaprasad@DA)─[~/Documents]
  └─$ less file4

└─(vidhyaprasad@DA)─[~/Documents]
  └─$ █
```

Output:

```
File Actions Edit View Help
first file
second file
third file
this is my second lab work
basic linux commands
this will show u the linux commands
my head command
file4 (END)█

File System
```

Syntax:

```
$less /etc/passwd
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
$ less /etc/passwd

└─(vidhyaprasad㉿DA)-[~/Documents]
$ █
```

Output:

```
File Actions Edit View Help
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/run/iodine:/usr/sbin/nologin
miredo:x:112:65534::/var/run/miredo:/usr/sbin/nologin
_rpc:x:113:65534::/run/rpcbind:/usr/sbin/nologin
usbmux:x:114:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tcpdump:x:115:121::/nonexistent:/usr/sbin/nologin
/etc/passwd█
```

7. cut:

The cut command is used for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field

Syntax:

- \$cut -b options filename

Syntax:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ cat file4
first file
second file
third file
this is my second lab work
basic linux commands
this will show u the linux commands
my head command

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ cut -b 1,2 file4
fi
se
th   Home
th
ba
th
my

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ cut -b 1,2,3 file4
fir
sec  1.22
thi
thi
bas
thi
my

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

8. paste:

It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output.

Syntax:

```
$ paste file1 file2
```

Example:

```
(vidhyaprasad@DA)-[~/Documents]
$ cat cutfile.txt
1
2
3
4
5     Home
6
7
8

(vidhyaprasad@DA)-[~/Documents]
$ paste cutfile.txt file4
1   first file
2   second file
3   .1 third file
4   this is my second lab work
5   basic linux commands
6   this will show u the linux commands
7   my head command
8

(vidhyaprasad@DA)-[~/Documents]
$
```

9. 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.

Syntax:

```
$uname
```

It will show you the actual username

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ uname
Linux
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

```
$uname -r
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ uname -r
5.10.0-kali7-amd64
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

```
$uname -v
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ uname -v
#1 SMP Debian 5.10.28-1kali1 (2021-04-12)
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

10. copy:

cp command is used to copy files from the current directory to a different directory.

Syntax:

```
$ cp filename directoryname
```

Example:

```
└─(vidhyaprasad㉿DA)-[~]
  └─$ ls
  Desktop Documents Downloads file1 file2 file3 Music nwlab Pictures Public Templates Videos
  └─(vidhyaprasad㉿DA)-[~]
  └─$ touch file5
  └─(vidhyaprasad㉿DA)-[~]
  └─$ ls
  Desktop Documents Downloads file1 file2 file3 file5 Music nwlab Pictures Public Templates Videos
  └─(vidhyaprasad㉿DA)-[~]
  └─$ ls nwlab
  file1 file2 file3 file4 linuxcmd mycommands1 mycommands1.txt mycommands.txt work1
  └─(vidhyaprasad㉿DA)-[~]
  └─$ cp file5 nwlab/
  └─(vidhyaprasad㉿DA)-[~]
  └─$ ls nwlab
  file1 file2 file3 file4 file5 linuxcmd mycommands1 mycommands1.txt mycommands.txt work1
  └─(vidhyaprasad㉿DA)-[~]
  └─$
```

11. mv:

The primary use of the mv command is to move files, it can also be used to rename files. The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory.

Syntax:

```
$ mv filename directoryname
```

Example:

```
(vidhyaprasad@DA)~]
$ ls
Desktop Documents Downloads file1 file2 file3 Music nwlab Pictures Public Templates Videos
(vidhyaprasad@DA)~]
$ touch file6

(vidhyaprasad@DA)~]
$ ls
Desktop Documents Downloads file1 file2 file3 file6 Music nwlab Pictures Public Templates Videos
Home
(vidhyaprasad@DA)~]
$ ls nwlab
file1 file2 file3 file4 file5 linuxcmd mycommands1 mycommands1.txt mycommands.txt work1

(vidhyaprasad@DA)~]
$ mv file6 nwlab/
(vidhyaprasad@DA)~]
$ ls nwlab
file1 file2 file3 file4 file5 linuxcmd mycommands1 mycommands1.txt mycommands.txt work1

(vidhyaprasad@DA)~]
$ ls
Desktop Documents Downloads file1 file2 file3 Music nwlab Pictures Public Templates Videos
(vidhyaprasad@DA)~]
```

12. locate:

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 caseinsensitive, so you can search for a file even if you don't remember its exact name. To search for a file that contains two or more words, use an asterisk (*).

Example:

```
root@kali:~# touch file1.txt
root@kali:~# locate file1.txt
/usr/lib/drredis/ruby/2.5.0/gems/rack-2.0.6/test/multipart/file1.txt
/usr/lib/drredis/ruby/2.5.0/gems/rubyzip-1.2.2/test/data/file1.txt
/usr/lib/drredis/ruby/2.5.0/gems/rubyzip-1.2.2/test/data/file1.txt.deflatedData
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/rack-1.6.11/test/m
ultipart/file1.txt
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/rubyzip-1.2.2/test
/data/file1.txt
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/rubyzip-1.2.2/test
/data/file1.txt.deflatedData
root@kali:~#
```

13. find:

Similar to the locate command, using find also searches for files and directories. • The difference is, you use the find command to locate files within a given directory. • As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its subdirectories

Syntax:

```
$ find /home/ -name filename
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
  └─$ find /home/ -name file5.txt
/home/vidhyaprasad/Documents/file5.txt

└─(vidhyaprasad㉿DA)-[~/Documents]
  └─$ █
```

14. 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. Usually output of a previous command is piped into the grep command.

Syntax:

```
$ Grep text filename
```

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ cat file4
first file
second file
third file
this is my second lab work
basic linux commands
this will show u the linux commands
my head command

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ grep basic file4
basic linux commands

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

Another Syntax:

\$grep word/etc/passwd

Example:

```
└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ grep vidhya /etc/passwd
vidhyaprasad:x:1000:1000:vidhyaprasad,,,:/home/vidhyaprasad:/usr/bin/zsh

└─(vidhyaprasad㉿DA)-[~/Documents]
└─$ █
```

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.

Syntax:

```
$ df -m
```

Example:

```
└─(vidhyaprasad㉿DA)─[~/nwlab]
$ df -m
Filesystem 1M-blocks Used Available Use% Mounted on
udev          588     0      588   0% /dev
tmpfs         125     1      124   1% /run
/dev/sda1    12037  9672     1733  85% /
tmpfs         623     0      623   0% /dev/shm
tmpfs          5     0       5   0% /run/lock
tmpfs         125     1      125   1% /run/user/1000
└─(vidhyaprasad㉿DA)─[~/nwlab]
$ █
```

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.

Syntax:

```
$ du -h
```

Example:

```
└─(vidhyaprasad㉿DA)─[~]
$ cd nwlab/
└─(vidhyaprasad㉿DA)─[~/nwlab]
$ du -h
4.0K  ./linuxcmd/cmd1
8.0K  ./linuxcmd
4.0K  ./work1
36K  .
└─(vidhyaprasad㉿DA)─[~/nwlab]
$ █
```

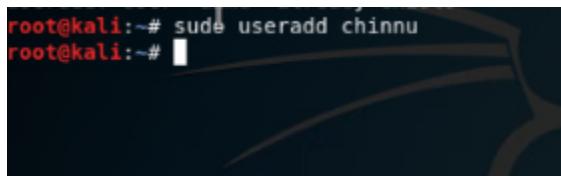
17. useradd :

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

Syntax:

```
$ sudo useradd name
```

Example:



```
root@kali:~# sudo useradd chinnu
```

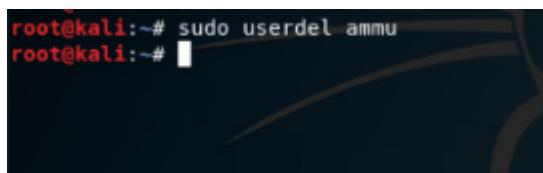
18. userdel

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

Syntax:

```
userdel UserName
```

Example:



```
root@kali:~# sudo userdel ammu
```

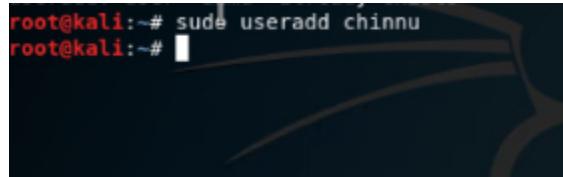
19. sudo

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.

Syntax:

```
sudo useradd name
```

Example:



```
root@kali:~# sudo useradd chinnu
```

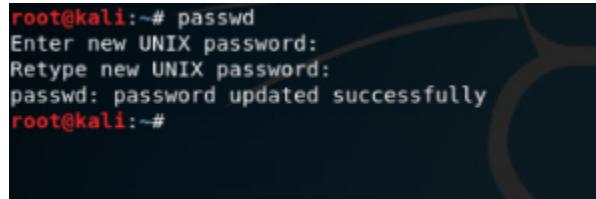
20. passwd

Changes passwords for user accounts. A normal user may only change the password for their own account, while the superuser may change the password for any account.

Syntax:

- passwd[option] [username]
- passwd
- passwd user1

Example:



```
root@kali:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali:~#
```

1. usermod

- usermod command is used to change the properties of a user in Linux through the command line
- command-line utility that allows you to modify a user's login information
 - #usermod –help
 - #usermod –u 2000 Tom

```
vidhya@DA:~$ usermod --help
Usage: usermod [options] LOGIN

Options:
  -b, --badnames          allow bad names
  -c, --comment COMMENT   new value of the GECOS field
  -d, --home HOME_DIR     new home directory for the user account
  -e, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE  set password inactive after expiration
                           to INACTIVE
  -g, --gid GROUP          force use GROUP as new primary group
  -G, --groups GROUPS      new list of supplementary GROUPS
  -a, --append               append the user to the supplemental GROUPS
                           mentioned by the -G option without removing
                           the user from other groups
  -h, --help                display this help message and exit
  -l, --login NEW_LOGIN    new value of the login name
  -L, --lock                  lock the user account
  -m, --move-home           move contents of the home directory to the
```

```
vidhya@DA:~$ usermod -u 2000 vidhya
usermod: user vidhya is currently used by process 46
vidhya@DA:~$
```

2. groupadd

- groupadd command creates a new group account using the values specified on the command line and the default values from the system.
- #groupadd student

```
vidhya@DA:~$ sudo groupadd student
[sudo] password for vidhya:
vidhya@DA:~$
```

3. groups

print the groups a user is in

- #groups alice

```
vidhya@DA:~$ sudo groupadd student
[sudo] password for vidhya:
vidhya@DA:~$ groups vidhya
vidhya : vidhya adm dialout cdrom floppy sudo audio dip video plugdev netdev
vidhya@DA:~$
```

4. groupdel

- groupdel command modifies the system account files, deleting all entries that refer to group. The named group must exist
- #groupdel marketing

```
vidhya@DA:~$ sudo groupdel student
vidhya@DA:~$
```

5. groupmod

- The groupmod command modifies the definition of the specified group by modifying the appropriate entry in the group database.

```
# groupmod -n group1 group2
```

```
vidhya@DA:~/mylinux$ sudo groupmod -n pug dog
vidhya@DA:~/mylinux$
```

6. chmod

- To change directory permissions of file/ Directory in Linux.

```
#chmod whowhatwhich file/directory
```

- chmod +rwx filename to add permissions.
- chmod -rwx directoryname to remove permissions.
- chmod +x filename to allow executable permissions.
- chmod -wx filename to take out write and executable permissions.

```
#chmod u+x test
```

```
#chmod g-rwx test
```

```
#chmod o-r test 4
```

```
vidhya@DA:~/mylinux$ chmod +rwx mywords.txt
vidhya@DA:~/mylinux$
```

```
vidhya@DA:~/mylinux$ mkdir books
vidhya@DA:~/mylinux$ ls -ld books
drwxr-xr-x 1 vidhya vidhya 4096 Aug 13 14:04 books
vidhya@DA:~/mylinux$ chmod g-w books
vidhya@DA:~/mylinux$ ls -ld books
drwxr-xr-x 1 vidhya vidhya 4096 Aug 13 14:04 books
vidhya@DA:~/mylinux$ chmod o-w books
vidhya@DA:~/mylinux$ ls -ld books
drwxr-xr-x 1 vidhya vidhya 4096 Aug 13 14:04 books
vidhya@DA:~/mylinux$ chmod o+w books
vidhya@DA:~/mylinux$ ls -ld books
drwxr-xrwx 1 vidhya vidhya 4096 Aug 13 14:04 books
vidhya@DA:~/mylinux$
```

7. chown

- The chown command allows you to change the user and/or group ownership of a given file, directory.

#chown Tom Test

```
vidhya@DA:~/mylinux$ sudo chown vidhya books
vidhya@DA:~/mylinux$ ls -ld sample
-rw-r--r-- 1 vidhya vidhya 0 Aug 13 14:14 sample
vidhya@DA:~/mylinux$
```

8. id

- id command in Linux is used to find out user and group names and numeric ID's (UID or group ID) of the current user.
- #id

```
vidhya@DA:~/mylinux$ id
uid=1000(vidhya) gid=1000(vidhya) groups=1000(vidhya),4(adm),20(dialout),24
(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),117(net
dev)
```

9. ps

- The ps command, short for Process Status, is a command line utility that is used to display or view information related to the processes running in a Linux system.
- PID – This is the unique process ID
- TTY – This is the type of terminal that the user is logged in to
- TIME – This is the time in minutes and seconds that the process has been running
- CMD – The command that launched the process
- #ps -a 5

```
vidhya@DA:~/mylinux$ ps
 PID TTY      TIME CMD
   9 tty1    00:00:01 bash
  98 tty1    00:00:00 ps
```

10. Top

- top command is used to show the Linux processes. It provides a dynamic real-time view of the running system
- #top -u rose

```
vidhya@DA:~/mylinux$ top -u vidhya
top - 14:21:47 up 33 min, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 4 total, 1 running, 3 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.1 us, 3.4 sy, 0.0 ni, 95.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 4012.9 total, 1633.9 free, 2155.0 used, 224.0 buff/cache
MiB Swap: 12288.0 total, 12251.6 free, 36.4 used. 1727.3 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  9 vidhya    20   0  17.7m  3.6m  3.5m S  0.0  0.1  0:01.40 bash
 100 vidhya   20   0  18.5m  2.2m  1.5m R  0.0  0.1  0:00.10 top
```

1. WC

- wc stands for word count.
- Used for counting purpose.
- It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.
- #wc state.txt

```
vidhya@DA:~/mylinux$ wc mywords.txt
4 10 58 mywords.txt
vidhya@DA:~/mylinux$ wc -l mywords.txt
4 mywords.txt
vidhya@DA:~/mylinux$ wc -w mywords.txt
10 mywords.txt
vidhya@DA:~/mylinux$ wc -c mywords.txt
58 mywords.txt
vidhya@DA:~/mylinux$ wc -m mywords.txt
58 mywords.txt
vidhya@DA:~/mylinux$
```

2. tar

- The Linux ‘tar’stands for tape archive, is used to create Archive and extract the Archive files
- Linux tar command to create compressed or uncompressed Archive files
- Options:
 - c : Creates Archive
 - x : Extract the archive
 - f : creates archive with given filename
 - t : displays or lists files in archived file
 - u : archives and adds to an existing archive file
 - v : Displays Verbose Information

- A : Concatenates the archive files
- z : zip, tells tar command that creates tar file using gzip
- j : filter archive tar file using tbzip
- W : Verify a archive file
- r : update or add file or directory in already existed .tar file

```
#tar cf archive.tar state.txt capital.txt //create archive file
#ls archive.tar #tar tf /archive.tar // list contents of tar archive file
```

- Extract an archive created with tar

```
#mkdir backup#cd backup
#tar xf /home/meera/Documents/Meera_Linux/archive.tar
```

```
vidhya@DA:~/mylinux$ ls
books  duplicate  jump  myclass.txt  mywords.txt  samp  sample
vidhya@DA:~/mylinux$ tar cf sample.tar myclass.txt jump samp
vidhya@DA:~/mylinux$ ls
books  duplicate  jump  myclass.txt  mywords.txt  samp  sample  sample.tar
vidhya@DA:~/mylinux$ tar xf /home/vidhya/mylinux/sample.tar
vidhya@DA:~/mylinux$ ls
books  duplicate  jump  myclass.txt  mywords.txt  samp  sample  sample.tar
vidhya@DA:~/mylinux$ tar czf sample.tar.gz /etc/
```

3. expr

- The expr command evaluates a given expression and displays its corresponding output. It is used for:
 - Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
 - Evaluating regular expressions, string operations like substring, length of strings etc.
 - Performing operations on variables inside a shell script

```
#expr 10 + 2
```

```
vidhya@DA:~/mylinux$ expr 10 + 20
30
vidhya@DA:~/mylinux$
```

4. Redirections & Piping

- A pipe is a form of redirection to send the output of one command/program/process to another command/program/process for further processing.
- Pipe is used to combine two or more commands, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on.

```
#ls -l | wc -l #cat /etc.passwd.txt | head -7 | tail -5
```

```
vidhya@DA:~/mylinux$ ls -l|wc -l
10
vidhya@DA:~/mylinux$
```

5. ssh

- ssh stands for “Secure Shell”.
- It is a protocol used to securely connect to a remote server/system.
- ssh is secure in the sense that it transfers the data in encrypted form between the host and the client.
- It transfers inputs from the client to the host and relays back the output. ssh runs at TCP/IP port 22.

```
#ssh user_name@host(IP/Domain_name) #ssh -X root@server1.example.com
```

```
vidhya@DA:~/mylinux$ ssh --help
unknown option -- -
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
vidhya@DA:~/mylinux$
```

6. scp

- SCP (secure copy) is a command-line utility that allows you to securely copy files and directories between two locations.
- With scp, you can copy a file or directory:
- From your local system to a remote system.
- From a remote system to your local system.
- Between two remote systems from your local system.
- Remote file system locations are specified in format [user@]host:/path

Syntax:

```
scp [OPTION] [user@]SRC_HOST:]file1[user@]DEST_HOST:]file2
```

```
$scp /etc/yum.config /etc/hosts ServerX:/home/student
```

```
$scp ServerX:/etc/hostname /home/student
```

7. ssh-keygen

- ssh-keygen command to generate a public/private authentication key pair. Authentication keys allow a user to connect to a remote system without supplying a password. Keys must be generated for each user separately. If you generate key pairs as the root user, only the root can use the keys.

```
$ssh-keygen -t rsa
```

```
vidhya@DA:~/mylinux$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vidhya/.ssh/id_rsa): rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in rsa
Your public key has been saved in rsa.pub
The key fingerprint is:
SHA256:j0cn6torHO+8tK5YDeiiTH7H9mNSG5m7pdOrEAvtUHU vidhya@DA
The key's randomart image is:
+--[RSA 3072]----+
|   . E          |
|   . .         |
|   .           |
|   o.          |
|  o.o. S o .  |
|  .+ +B = o   |
|  .. o=ooX.o   |
|  +. o B+@++   |
|  .o. + *#@+. |
+---[SHA256]----+
vidhya@DA:~/mylinux$
```

8. ssh-copy-id

- The ssh-copy-id command allows you to install an SSH key on a remote server's authorized keys.
- This command facilitates SSH key login, which removes the need for a password for each login, thus ensuring a password-less, automatic login process.

\$ssh-copy-id username@remote_host

1. Managing Files, Creating Users and Groups Using Command-line tools

1. a. Create six files with name of the form songX.mp3

Syntax:

```
$ touch song1.mp3 song2.mp3 song3.mp3 song4.mp3 song5.mp3 song6.mp3
```

```
vidhya@DA:~/mylinux$ cd mylinux
vidhya@DA:~/mylinux$ touch song1.mp3 song2.mp3 song3.mp3 song4.mp3 song5.mp3 song6.mp3
vidhya@DA:~/mylinux$ ls
books      jump      mywords.txt  rsa.pub  sample      sample.tar.gz  song2.mp3  song4.mp3  song6.mp3
duplicate  myclass.txt  rsa        samp     sample.tar    song1.mp3    song3.mp3  song5.mp3
vidhya@DA:~/mylinux$
```

1. b. Create six files with name of the form snapX.mp3

Syntax:

```
$ touch snap1.mp3 snap2.mp3 snap3.mp3 snap4.mp3 snap5.mp3 snap6.mp3
```

```
vidhya@DA:~/mylinux$ touch snap1.mp3 snap2.mp3 snap3.mp3 snap4.mp3 snap5.mp3 snap6.mp3
vidhya@DA:~/mylinux$ ls
books      myclass.txt  rsa.pub  sample.tar      snap2.mp3  snap5.mp3  song2.mp3  song5.mp3
duplicate  mywords.txt  samp    sample.tar.gz    snap3.mp3  snap6.mp3  song3.mp3  song6.mp3
jump      rsa          sample   snap1.mp3      snap4.mp3  song1.mp3  song4.mp3
vidhya@DA:~/mylinux$
```

1. c. Create six files with name of the form filmX.mp3 (In each set, replace X with the numbers 1 through 6)

Syntax:

```
$ touch film1.mp3 film2.mp3 film3.mp3 film4.mp3 film5.mp3 film6.mp3
```

```
vidhya@DA:~/mylinux$ touch film1.mp3 film2.mp3 film3.mp3 film4.mp3 film5.mp3 film6.mp3
vidhya@DA:~/mylinux$ ls
books      film2.mp3  film5.mp3  myclass.txt  rsa.pub  sample.tar      snap2.mp3  snap5.mp3  song2.mp3  song5.mp3
duplicate  film3.mp3  film6.mp3  mywords.txt  samp    sample.tar.gz    snap3.mp3  snap6.mp3  song3.mp3  song6.mp3
film1.mp3  film4.mp3  jump      rsa        sample   snap1.mp3      snap4.mp3  song1.mp3  song4.mp3
vidhya@DA:~/mylinux$
```

- From your home directory, move the song files into your music subdirectory, the snapshot files into your pictures subdirectory, and the movie files into videos subdirectory.

Syntax :

Music subdirectory;

```
$ mv song2.mp3 song4.mp3 song5.mp3 song6.mp3 song3.mp3 music/
```

```
vidhya@DA:~/mylinux$ ls
books  film2.mp3  film5.mp3  music      rsa      sample      snap1.mp3  snap4.mp3  song1.mp3  song4.mp3
duplicate  film3.mp3  film6.mp3  myclass.txt  rsa.pub  sample.tar  snap2.mp3  snap5.mp3  song2.mp3  song5.mp3
film1.mp3  film4.mp3  jump      mywords.txt  samp    sample.tar.gz  snap3.mp3  snap6.mp3  song3.mp3  song6.mp3
vidhya@DA:~/mylinux$ mv song1.mp3 music/
vidhya@DA:~/mylinux$ mv song2.mp3 song4.mp3 song5.mp3 song6.mp3 song3.mp3 music/
vidhya@DA:~/mylinux$ cd music
vidhya@DA:~/mylinux/music$ ls
song1.mp3  song2.mp3  song3.mp3  song4.mp3  song5.mp3  song6.mp3
vidhya@DA:~/mylinux/music$
```

Syntax :

Picture subdirectory;

```
$ mv snap2.mp3 snap1.mp3 snap4.mp3 snap5.mp3 snap6.mp3 snap3.mp3 picture/
```

```
vidhya@DA:~/mylinux$ mkdir picture
vidhya@DA:~/mylinux$ mv snap2.mp3 snap1.mp3 snap4.mp3 snap5.mp3 snap6.mp3 snap3.mp3 picture/
vidhya@DA:~/mylinux$ cd picture
vidhya@DA:~/mylinux/picture$ ls
snap1.mp3  snap2.mp3  snap3.mp3  snap4.mp3  snap5.mp3  snap6.mp3
vidhya@DA:~/mylinux/picture$
```

Syntax :

Videos subdirectory;

```
$ mv film2.mp3 film1.mp3 film4.mp3 film5.mp3 film6.mp3 film3.mp3 videos/
```

```
vidhya@DA:~/mylinux/picture$ cd ..
vidhya@DA:~/mylinux$ mkdir videos
vidhya@DA:~/mylinux$ mv film2.mp3 film1.mp3 film4.mp3 film5.mp3 film6.mp3 film3.mp3 videos/
vidhya@DA:~/mylinux$ cd videos
vidhya@DA:~/mylinux/videos$ ls
film1.mp3  film2.mp3  film3.mp3  film4.mp3  film5.mp3  film6.mp3
vidhya@DA:~/mylinux/videos$
```

3. In your home directory, create three subdirectories for organizing your files. Call these directories friends, family, and work. Create all three with one command.

Syntax :

```
$ mkdir friends family work
```

```
vidhya@DA:~/mylinux$ mkdir friends family work
vidhya@DA:~/mylinux$ ls
books      family   jump   myclass.txt  picture  rsa.pub  sample    sample.tar.gz  work
duplicate  friends  music  mywords.txt  rsa       samp    sample.tar  videos
vidhya@DA:~/mylinux$
```

4. Copy song files to the friends folder and snap files to family folder

Friends folder Syntax :

```
$ cp song2.mp3 song4.mp3 song5.mp3 song6.mp3 song3.mp3
/home/vidhya/mylinux/friends/
```

```
vidhya@DA:~/mylinux/music$ cp song2.mp3 song4.mp3 song5.mp3 song6.mp3 song3.mp3 /home/vidhya/mylinux/friends/
vidhya@DA:~/mylinux/music$ ls
song1.mp3  song2.mp3  song3.mp3  song4.mp3  song5.mp3  song6.mp3
vidhya@DA:~/mylinux$ cd ..
vidhya@DA:~/mylinux$ ls
books      family   jump   myclass.txt  picture  rsa.pub  sample    sample.tar.gz  work
duplicate  friends  music  mywords.txt  rsa       samp    sample.tar  videos
vidhya@DA:~/mylinux$ cd friends
vidhya@DA:~/mylinux/friends$ ls
song2.mp3  song3.mp3  song4.mp3  song5.mp3  song6.mp3
vidhya@DA:~/mylinux/friends$
```

Family folder Syntax :

```
$ cp snap2.mp3 snap1.mp3 snap4.mp3 snap5.mp3 snap6.mp3 snap3.mp3
/home/vidhya/mylinux/family/
```

```
vidhya@DA:~/mylinux$ cd picture
vidhya@DA:~/mylinux/picture$ ls
snap1.mp3  snap2.mp3  snap3.mp3  snap4.mp3  snap5.mp3  snap6.mp3
vidhya@DA:~/mylinux/picture$ cp snap2.mp3 snap1.mp3 snap4.mp3 snap5.mp3 snap6.mp3 snap3.mp3 /home/vidhya/mylinux/family/
vidhya@DA:~/mylinux/picture$ cd family
-bash: cd: family: No such file or directory
vidhya@DA:~/mylinux/picture$ cd ..
vidhya@DA:~/mylinux$ cd family
vidhya@DA:~/mylinux/family$ ls
snap1.mp3  snap2.mp3  snap3.mp3  snap4.mp3  snap5.mp3  snap6.mp3
vidhya@DA:~/mylinux/family$
```

5. Attempt to delete both family and friends projects with a single rmdir command.

Syntax :

```
$ rmdir {friends,family}
```

```
vidhya@DA:~/mylinux$ rmdir {friends,family}
rmdir: failed to remove 'friends': Directory not empty
rmdir: failed to remove 'family': Directory not empty
vidhya@DA:~/mylinux$
```

6. Use another command that will succeed in deleting both the family and friends folder.

Syntax :

```
rm -r friends family
```

```
vidhya@DA:~/mylinux$ rm -r friends family
vidhya@DA:~/mylinux$ ls
books  duplicate  jump  music  myclass.txt  mywords.txt  picture  rsa  rsa.pub  samp  sample  sample.tar  sample.tar.gz  videos  work
vidhya@DA:~/mylinux$
```

7. Redirect a long listing of all home directory files, including hidden, into a file named allfiles.txt. Confirm that the file contains the listing.

Syntax :

```
$ ls -a > allfiles.txt
```

```
vidhya@DA:~/mylinux$ ls -a > allfiles.txt
vidhya@DA:~/mylinux$ ls
allfiles.txt  duplicate  music      mywords.txt  rsa        samp     sample.tar    videos
books        jump       myclass.txt  picture     rsa.pub    sample   sample.tar.gz  work
```

8. In the command window, display today's date with day of the week, month, date and year

Syntax :

```
$ date
```

```
vidhya@DA:~/mylinux$ date
Tue Aug 17 19:26:24 IST 2021
vidhya@DA:~/mylinux$
```

9. Add the user Juliet

Syntax :

```
$ useradd username
```

```
vidhya@DA:~/mylinux$ sudo useradd juliet  
[sudo] password for vidhya:
```

10. Confirm that Juliet has been added by examining the /etc/passwd file

Syntax :

```
$ cat /etc/passwd |grep juliet
```

```
vidhya@DA:~/mylinux$ cat /etc/passwd |grep juliet  
juliet:x:1001:1003::/home/juliet:/bin/sh  
vidhya@DA:~/mylinux$
```

11. Use the passwd command to initialize Juliet's password

Syntax :

```
$ sudo passwd juliet
```

```
vidhya@DA:~/mylinux$ sudo passwd juliet  
New password:  
Retype new password:  
passwd: password updated successfully  
vidhya@DA:~/mylinux$
```

12. Create a supplementary group called Shakespeare with a group id of 30000.

Syntax :

```
$ sudo groupadd -g 30000 shakespeare
```

```
vidhya@DA:~/mylinux$ sudo groupadd -g 30000 shakespeare  
vidhya@DA:~/mylinux$ ls
```

13. Create a supplementary group called artists.

Syntax:

```
$ sudo groupadd artist
```

```
vidhya@DA:~/mylinux$ sudo groupadd artist
```

14. Confirm that Shakespeare and artists have been added by examining the /etc/group file.

Syntax:

```
$ less /etc/group
```

```
pug:x:1002:  
juliet:x:1003:  
shakespeare:x:30000:  
artist:x:30001:  
(END)
```

15. Add the Juliet user to the Shakespeare group as a supplementary group.

Syntax :

```
$ sudo usermod -g shakespeare juliet
```

```
vidhya@DA:~/mylinux$ sudo usermod -g shakespeare juliet  
vidhya@DA:~/mylinux$ less /etc/group
```

16. Confirm that Juliet has been added using the id command.

Syntax :

```
$ id juliet
```

```
vidhya@DA:~/mylinux$ id juliet  
uid=1001(juliet) gid=30000(shakespeare) groups=30000(shakespeare)  
vidhya@DA:~/mylinux$
```

17. Add Romeo and Hamlet to the Shakespeare group.

Syntax :

```
$ sudo useradd romeo  
$ sudo useradd hamlet  
$ sudo usermod -g shakespeare romeo  
$ sudo usermod -g shakespeare hamlet
```

```
vidhya@DA:~/mylinux$ sudo useradd romeo  
vidhya@DA:~/mylinux$ sudo useradd hamlet  
vidhya@DA:~/mylinux$ sudo usermod -g shakespeare romeo  
vidhya@DA:~/mylinux$ sudo usermod -g shakespeare hamlet  
vidhya@DA:~/mylinux$ less /etc/group  
vidhya@DA:~/mylinux$
```

18. Add Reba, Dolly and Elvis to the artists group.

Syntax:

```
sudo useradd reba  
$ sudo useradd dolly  
$ sudo useradd elvis  
$ sudo usermod -g artist reba  
$ sudo usermod -g artist dolly  
$ sudo usermod -g artist elvis
```

```
vidhya@DA:~/mylinux$ sudo useradd reba  
vidhya@DA:~/mylinux$ sudo useradd dolly  
vidhya@DA:~/mylinux$ sudo useradd elvis  
vidhya@DA:~/mylinux$ sudo usermod -g artist reba  
vidhya@DA:~/mylinux$ sudo usermod -g artist dolly  
vidhya@DA:~/mylinux$ sudo usermod -g artist elvis  
vidhya@DA:~/mylinux$
```

19. Verify the supplemental group memberships by examining the /etc/group file

Syntax:

```
$ less /etc/group
```

```
shakespear@DA:~$ less /etc/group
artist:x:30001:
romeo:x:30002:
hamlet:x:30003:
reba:x:1004:
dolly:x:1005:
elvis:x:1006:
(END)
```

20. Attempt to remove user Dolly.

Syntax:

```
$ sudo userdel dolly
```

```
vidhya@DA:~/mylinux$ sudo userdel dolly
```

1. Ping & traceroute tests

Ping and Trace Route tests can help to identify any connection issues between your network and a specified server (or website) address.

PING test

The PING command is used to test the connection and latency between two network connections. The PING command sends packets of information to a specified IP Address and then measures the time it takes to get a response from the specified computer or device.

```
C:\Users\DA>ping google.com

Pinging google.com [2404:6800:4002:820::200e] with 32 bytes of data:
Reply from 2404:6800:4002:820::200e: time=157ms
Reply from 2404:6800:4002:820::200e: time=253ms
Reply from 2404:6800:4002:820::200e: time=160ms
Reply from 2404:6800:4002:820::200e: time=108ms

Ping statistics for 2404:6800:4002:820::200e:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 108ms, Maximum = 253ms, Average = 169ms

C:\Users\DA>ping -a google.com

Pinging google.com [2404:6800:4002:820::200e] with 32 bytes of data:
Reply from 2404:6800:4002:820::200e: time=1297ms
Reply from 2404:6800:4002:820::200e: time=186ms
Reply from 2404:6800:4002:820::200e: time=186ms
Reply from 2404:6800:4002:820::200e: time=187ms

Ping statistics for 2404:6800:4002:820::200e:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 186ms, Maximum = 1297ms, Average = 464ms

C:\Users\DA>ping -j google.com

Pinging google.com [142.250.195.14] with 32 bytes of data:
General failure.
General failure.
General failure.
General failure.

Ping statistics for 142.250.195.14:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\DA>ping -4 google.com

Pinging google.com [142.250.195.14] with 32 bytes of data:
Reply from 142.250.195.14: bytes=32 time=143ms TTL=113
```

Trace Route test

The TRACERT command is used to conduct a similar test to PING, but instead of displaying the time it takes to connect, it looks at the exact server hops required to connect your computer to the server.

You should already have the CMD prompt dialogue box open, after performing the PING test above.

```
C:\Users\DA>tracert -d www.google.com

Tracing route to www.google.com [2404:6800:4007:827::2004]
over a maximum of 30 hops:

 1    45 ms    97 ms    97 ms  2409:4073:4e01:18b8::58
 2    *          *          * Request timed out.
 3   183 ms    97 ms    97 ms  2405:200:365:eeee:20::344
 4    81 ms    42 ms    45 ms  2405:200:801:1100::3da
 5   147 ms    54 ms    46 ms  2405:200:801:1100::3d9
 6   191 ms    94 ms    49 ms  2405:200:100::1:0:37e
 7   146 ms    64 ms    63 ms  2001:4860:1:1::171
 8   158 ms   112 ms    68 ms  2001:4860:1:1::8f2
 9   137 ms    56 ms    64 ms  2404:6800:8133::1
10   183 ms    62 ms    67 ms  2001:4860:0:1::55d0
11   181 ms    54 ms    65 ms  2001:4860:0:1::559d
12   158 ms    55 ms    76 ms  2404:6800:4007:827::2004

Trace complete.

C:\Users\DA>
C:\Users\DA>tracert www.google.com

Tracing route to www.google.com [2404:6800:4007:827::2004]
over a maximum of 30 hops:

 1    57 ms    97 ms    96 ms  2409:4073:4e01:18b8::58
 2    *          *          * Request timed out.
 3   123 ms    97 ms    97 ms  2405:200:365:eeee:20::344
 4    82 ms    55 ms    85 ms  2405:200:801:1100::3da
 5    55 ms    83 ms    56 ms  2405:200:801:1100::3d9
 6   150 ms    62 ms    66 ms  2405:200:100::1:0:37e
 7   177 ms    96 ms    52 ms  2001:4860:1:1::171
 8    67 ms    55 ms    62 ms  2001:4860:1:1::8f2
 9   118 ms    63 ms    57 ms  2404:6800:8133::1
10   127 ms    54 ms    64 ms  2001:4860:0:1::55d0
11    69 ms    54 ms    67 ms  2001:4860:0:1::559d
12   177 ms    54 ms    65 ms  maa03s42-in-x04.1e100.net [2404:6800:4007:827::2004]

Trace complete.
```

```
C:\Users\DA>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

 1      4 ms      4 ms      4 ms  192.168.43.199
 2      *          *          *      Request timed out.
 3    112 ms     97 ms     98 ms  56.8.126.61
 4     70 ms     43 ms     44 ms  172.26.104.197
 5   184 ms     97 ms     97 ms  172.26.104.211
 6   168 ms     97 ms     97 ms  192.168.14.38
 7     80 ms     44 ms     44 ms  192.168.14.37
 8   116 ms     97 ms     97 ms  172.16.3.14
 9   135 ms     56 ms     35 ms  172.16.81.0
10   110 ms     44 ms     56 ms  172.16.0.159
11   123 ms     98 ms     97 ms  172.16.3.15
12   104 ms     97 ms     97 ms  172.16.5.70

C:\Users\DA>route print *157
=====
Interface List
 2...b4 b6 86 0d 60 71 .....Realtek PCIe GBE Family Controller
 12...0a 00 27 00 00 0c .....VirtualBox Host-Only Ethernet Adapter
 18...82 c5 f2 7c 3b 59 .....Microsoft Wi-Fi Direct Virtual Adapter
 8...80 c5 f2 7c 3b 59 .....Microsoft Wi-Fi Direct Virtual Adapter #2
 10...80 c5 f2 7c 3b 59 .....Realtek RTL8723DE 802.11b/g/n PCIe Adapter
 7...80 c5 f2 7c 3b 58 .....Bluetooth Device (Personal Area Network)
 1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
  None
Persistent Routes:
  None

IPv6 Route Table
=====
Active Routes:
  None
Persistent Routes:
  None

C:\Users\DA>
```

```
C:\Users\DA>route -6

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
          [MASK netmask] [gateway] [METRIC metric] [IF interface]

-f           Clears the routing tables of all gateway entries. If this is
            used in conjunction with one of the commands, the tables are
            cleared prior to running the command.

-p           When used with the ADD command, makes a route persistent across
            boots of the system. By default, routes are not preserved
            when the system is restarted. Ignored for all other commands,
            which always affect the appropriate persistent routes.

-4           Force using IPv4.

-6           Force using IPv6.

command      One of these:
              PRINT    Prints a route
              ADD     Adds a route
              DELETE   Deletes a route
              CHANGE   Modifies an existing route

destination   Specifies the host.
MASK         Specifies that the next parameter is the 'netmask' value.
netmask       Specifies a subnet mask value for this route entry.
              If not specified, it defaults to 255.255.255.255.

gateway       Specifies gateway.

interface     the interface number for the specified route.

METRIC       specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.
```

```
C:\Users\DA>route print -4
=====
Interface List
 2...b4 b6 86 0d 60 71 ....Realtek PCIe GBE Family Controller
 12...0a 00 27 00 00 0c ....VirtualBox Host-Only Ethernet Adapter
 18...82 c5 f2 7c 3b 59 ....Microsoft Wi-Fi Direct Virtual Adapter
  8...80 c5 f2 7c 3b 59 ....Microsoft Wi-Fi Direct Virtual Adapter #2
 10...80 c5 f2 7c 3b 59 ....Realtek RTL8723DE 802.11b/g/n PCIe Adapter
  7...80 c5 f2 7c 3b 58 ....Bluetooth Device (Personal Area Network)
  1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination      Netmask        Gateway       Interface Metric
          0.0.0.0        0.0.0.0    192.168.43.199  192.168.43.31    55
          127.0.0.0      255.0.0.0        On-link      127.0.0.1    331
          127.0.0.1      255.255.255.255  On-link      127.0.0.1    331
 127.255.255.255      255.255.255.255  On-link      127.0.0.1    331
          169.254.0.0      255.255.0.0        On-link    192.168.56.1    30
 169.254.255.255      255.255.255.255  On-link    192.168.56.1   281
          192.168.43.0      255.255.255.0        On-link  192.168.43.31   311
          192.168.43.31      255.255.255.255  On-link  192.168.43.31   311
 192.168.43.255      255.255.255.255  On-link  192.168.43.31   311
          192.168.56.0      255.255.255.0        On-link  192.168.56.1   281
          192.168.56.1      255.255.255.255  On-link  192.168.56.1   281
 192.168.56.255      255.255.255.255  On-link  192.168.56.1   281
          224.0.0.0        240.0.0.0        On-link      127.0.0.1    331
          224.0.0.0        240.0.0.0        On-link    192.168.56.1   281
          224.0.0.0        240.0.0.0        On-link  192.168.43.31   311
 255.255.255.255      255.255.255.255  On-link      127.0.0.1    331
 255.255.255.255      255.255.255.255  On-link    192.168.56.1   281
 255.255.255.255      255.255.255.255  On-link  192.168.43.31   311
=====

Persistent Routes:
  None
```

```
      224.0.0.0      240.0.0.0      On-link      192.168.43.31      311
 255.255.255.255 255.255.255.255  On-link          127.0.0.1      331
 255.255.255.255 255.255.255.255  On-link          192.168.56.1      281
 255.255.255.255 255.255.255.255  On-link          192.168.43.31      311
=====
Persistent Routes:
  None

IPv6 Route Table
=====
Active Routes:
 If Metric Network Destination      Gateway
 10      71  ::/0                  fe80::560e:2dff:fea5:352d
   1      331  ::1/128             On-link
 10      71  2409:4073:4e01:18b8::/64  On-link
 10      311  2409:4073:4e01:18b8:fca3:b447:9b86:7707/128
                                         On-link
 10      311  2409:4073:4e01:18b8:fdee:575c:b600:3049/128
                                         On-link
 12      281  fe80::/64            On-link
 10      311  fe80::/64            On-link
 12      281  fe80::1d38:6d43:d631:edb4/128
                                         On-link
 10      311  fe80::fca3:b447:9b86:7707/128
                                         On-link
   1      331  ff00::/8            On-link
 12      281  ff00::/8            On-link
 10      311  ff00::/8            On-link
=====
Persistent Routes:
  None

C:\Users\DA>
```

```
Minimum: 125ms, Maximum: 211ms, Average: 175ms
C:\Users\DA>route print
=====
Interface List
 2...b4 b6 86 0d 60 71 .... Realtek PCIe GBE Family Controller
 12...0a 00 27 00 00 0c .... VirtualBox Host-Only Ethernet Adapter
 18...82 c5 f2 7c 3b 59 .... Microsoft Wi-Fi Direct Virtual Adapter
  8...80 c5 f2 7c 3b 59 .... Microsoft Wi-Fi Direct Virtual Adapter #2
 10...80 c5 f2 7c 3b 59 .... Realtek RTL8723DE 802.11b/g/n PCIe Adapter
  7...80 c5 f2 7c 3b 58 .... Bluetooth Device (Personal Area Network)
  1..... Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination      Netmask        Gateway        Interface Metric
          0.0.0.0      0.0.0.0    192.168.43.199  192.168.43.31    55
         127.0.0.0    255.0.0.0        On-link       127.0.0.1    331
         127.0.0.1  255.255.255.255        On-link       127.0.0.1    331
 127.255.255.255  255.255.255.255        On-link       127.0.0.1    331
         169.254.0.0  255.255.0.0        On-link     192.168.56.1    30
 169.254.255.255  255.255.255.255        On-link     192.168.56.1   281
         192.168.43.0  255.255.255.0        On-link     192.168.43.31   311
         192.168.43.31  255.255.255.255        On-link     192.168.43.31   311
 192.168.43.255  255.255.255.255        On-link     192.168.43.31   311
         192.168.56.0  255.255.255.0        On-link     192.168.56.1   281
         192.168.56.1  255.255.255.255        On-link     192.168.56.1   281
 192.168.56.255  255.255.255.255        On-link     192.168.56.1   281
         224.0.0.0    240.0.0.0        On-link       127.0.0.1    331
         224.0.0.0    240.0.0.0        On-link     192.168.56.1   281
         224.0.0.0    240.0.0.0        On-link     192.168.43.31   311

C:\Users\DA>tracert 22.110.0.1
Tracing route to 22.110.0.1 over a maximum of 30 hops

 1     8 ms      7 ms      2 ms  192.168.43.199
 2     *          *          * Request timed out.
 3  135 ms     97 ms     97 ms  56.8.126.45
 4  194 ms     96 ms     97 ms  172.26.104.197
 5  109 ms     41 ms     52 ms  172.26.104.211
 6  112 ms     50 ms     66 ms  192.168.14.32
 7  157 ms    128 ms    166 ms  192.168.14.35
 8  124 ms    198 ms     97 ms  172.16.3.14
 9  182 ms     97 ms    199 ms  172.16.81.2
10  117 ms     61 ms     74 ms  172.16.0.159
11  191 ms    199 ms    200 ms  172.16.3.15
12  105 ms    115 ms    119 ms  172.16.2.60
13  395 ms    302 ms    302 ms  103.198.140.106
14  615 ms    301 ms    302 ms  103.198.140.54
```

2. Nslookup

Microsoft Windows includes a tool called NSLOOKUP that you can use via the command prompt. This tool can be used to check DNS records propagation and resolution using different servers, and perform other troubleshooting steps.

```
C:\Users\DA>nslookup
Default Server: UnKnown
Address: 192.168.43.10

>
C:\Users\DA>nslookup -g=MX google.com
*** Invalid option: g=MX
Server: UnKnown
Address: 192.168.43.10

Non-authoritative answer:
Name: google.com
Addresses: 2404:6800:4007:81d::200e
           142.250.194.46

C:\Users\DA>
```

3. Netstat

On Windows 10, netstat (network statistics) has been around for a long time, and it's a command-line tool that you can use in Command Prompt to display statistics for all network connections. It allows you to understand open and connected ports to monitor and troubleshoot networking problems for system or applications.

```
C:\Users\DA>netstat
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.43.31:63272   20.197.71.89:https    ESTABLISHED
  TCP    192.168.43.31:63338   20.44.229.112:https  TIME_WAIT
  TCP    192.168.43.31:63339   20.44.229.112:https  ESTABLISHED
  TCP    [2409:4073:4e01:18b8:a094:fcde:e73c:2dc5]:63333  [2606:2800:147:120f:30c:1ba0:fc6:265a]:https  CLOSE_WAIT

C:\Users\DA>
```

netstat -n

command to display active connections showing numeric IP address and port number instead of trying to determine the names .

netstat -n INTERVAL

In the command, make sure to replace INTERVAL for the number (in seconds) you want to redisplay the information.

```
C:\Users\DA>netstat -n

Active Connections

Proto Local Address          Foreign Address        State
TCP   192.168.43.31:63272    20.197.71.89:443    ESTABLISHED
TCP   192.168.43.31:63338    20.44.229.112:443   TIME_WAIT
TCP   192.168.43.31:63339    20.44.229.112:443   TIME_WAIT
TCP   192.168.43.31:63340    20.44.229.112:443   ESTABLISHED
TCP   [2409:4073:4e01:18b8:a094:fcde:e73c:2dc5]:63333  [2606:2800:147:120f:30c:1ba0:fc6:265a]:443  CLOSE_WAIT

C:\Users\DA>netstat -n 5

Active Connections

Proto Local Address          Foreign Address        State
TCP   192.168.43.31:63272    20.197.71.89:443    ESTABLISHED
TCP   192.168.43.31:63339    20.44.229.112:443   TIME_WAIT
TCP   192.168.43.31:63340    20.44.229.112:443   ESTABLISHED
TCP   [2409:4073:4e01:18b8:a094:fcde:e73c:2dc5]:63333  [2606:2800:147:120f:30c:1ba0:fc6:265a]:443  CLOSE_WAIT

Active Connections
```

netstat -a

The netstat -a command displays all active and inactive connections, and the TCP and UDP ports the device is currently listening.

```
C:\Users\DA>netstat -a

Active Connections

Proto Local Address          Foreign Address        State
TCP   0.0.0.0:135             DA:0                LISTENING
TCP   0.0.0.0:445             DA:0                LISTENING
TCP   0.0.0.0:5040            DA:0                LISTENING
TCP   0.0.0.0:5357            DA:0                LISTENING
TCP   0.0.0.0:49664           DA:0                LISTENING
TCP   0.0.0.0:49665           DA:0                LISTENING
TCP   0.0.0.0:49666           DA:0                LISTENING
TCP   0.0.0.0:49667           DA:0                LISTENING
TCP   0.0.0.0:49668           DA:0                LISTENING
TCP   0.0.0.0:49669           DA:0                LISTENING
TCP   127.0.0.1:5354          DA:0                LISTENING
TCP   192.168.43.31:139       DA:0                LISTENING
TCP   192.168.43.31:63272    20.197.71.89:https  ESTABLISHED
TCP   192.168.43.31:63339    20.44.229.112:https  TIME_WAIT
TCP   192.168.43.31:63340    20.44.229.112:https  ESTABLISHED
TCP   192.168.43.31:63341    20.44.229.112:https  ESTABLISHED
TCP   192.168.56.1:139        DA:0                LISTENING
TCP   [::]:135                DA:0                LISTENING
TCP   [::]:445                DA:0                LISTENING
TCP   [::]:5357                DA:0                LISTENING
TCP   [::]:49664               DA:0                LISTENING
TCP   [::]:49665               DA:0                LISTENING
TCP   [::]:49666               DA:0                LISTENING
TCP   [::]:49667               DA:0                LISTENING
TCP   [::]:49668               DA:0                LISTENING
TCP   [::]:49669               DA:0                LISTENING
TCP   [2409:4073:4e01:18b8:a094:fcde:e73c:2dc5]:63333  [2606:2800:147:120f:30c:1ba0:fc6:265a]:https  CLOSE_WAIT
UDP  0.0.0.0:123              *:*
```

netstat -b

The netstat -b command lists all the executables (applications) associated with each connection. Sometimes, applications may open multiple connections.

```
C:\Users\DA>netstat -b  
The requested operation requires elevation.
```

netstat -e

The netstat -e command generates a statistic of the network interface, which shows information like the number of bytes, unicast and non-unicast sent and received packets. You can also see discarded packets and errors and unknown protocols, which can you troubleshoot networking problems.

```
C:\Users\DA>netstat -e  
Interface Statistics  
  
          Received          Sent  
  
Bytes          108066      502866  
Unicast packets      511        448  
Non-unicast packets      0        3780  
Discards          0          0  
Errors            0          0  
Unknown protocols      0          0  
  
C:\Users\DA>
```

4. ipconfig

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, ipconfig displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

PARAMETERS:

/all: Displays the full TCP/IP configuration for all adapters. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.

/displaydns: Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer. The DNS Client service uses this information to resolve frequently queried names quickly, before querying its configured DNS servers.

/flushdns: Flushes and resets the contents of the DNS client resolver cache. During DNS troubleshooting, you can use this procedure to discard negative cache entries from the cache, as well as any other entries that have been added dynamically.

/registerdns: Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer. You can use this parameter to troubleshoot a failed DNS name registration or resolve a dynamic update problem between a client and the DNS server without rebooting the client computer. The DNS settings in the advanced properties of the TCP/IP protocol determine which names are registered in DNS.

```
C:\Users\DA>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 2:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Ethernet adapter VirtualBox Host-Only Network:

  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::1d38:6d43:d631:edb4%12
  IPv4 Address. . . . . : 192.168.56.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 2:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 1:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . :
  IPv6 Address. . . . . : 2409:4073:4e01:18b8:fca3:b447:9b86:7707
  Temporary IPv6 Address. . . . . : 2409:4073:4e01:18b8:a094:fcde:e73c:2dc5
  Link-local IPv6 Address . . . . . : fe80::fca3:b447:9b86:7707%10
  IPv4 Address. . . . . : 192.168.43.31
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::560e:2dff:fea5:352d%10
                                         192.168.43.10

Ethernet adapter Bluetooth Network Connection:
```

```
C:\Users\DA>ipconfig /displaying

Error: unrecognized or incomplete command line.

USAGE:
    ipconfig [/allcompartments] [/? | /all |
                                /renew [adapter] | /release [adapter] |
                                /renew6 [adapter] | /release6 [adapter] |
                                /flushdns | /displaydns | /registerdns |
                                /showclassid adapter |
                                /setclassid adapter [classid] |
                                /showclassid6 adapter |
                                /setclassid6 adapter [classid] ]

where
    adapter           Connection name
                      (wildcard characters * and ? allowed, see examples)

Options:
    /?                Display this help message
    /all              Display full configuration information.
    /release          Release the IPv4 address for the specified adapter.
    /release6         Release the IPv6 address for the specified adapter.
    /renew            Renew the IPv4 address for the specified adapter.
    /renew6           Renew the IPv6 address for the specified adapter.
    /flushdns         Purges the DNS Resolver cache.
    /registerdns     Refreshes all DHCP leases and re-registers DNS names
    /displaydns      Display the contents of the DNS Resolver Cache.
    /showclassid     Displays all the dhcp class IDs allowed for adapter.
    /setclassid      Modifies the dhcp class id.
    /showclassid6    Displays all the IPv6 DHCP class IDs allowed for adapter.
    /setclassid6    Modifies the IPv6 DHCP class id.

The default is to display only the IP address, subnet mask and
default gateway for each adapter bound to TCP/IP.
```

Other Networking Commands

1. Hostname Command

A very simple command that displays the host name of your machine. This is much quicker than going to the control panel>system route.

```
C:\Users\DA>hostname  
DA  
C:\Users\DA>
```

2. getmac Command

Another very simple command that shows the MAC address of your network interfaces

```
C:\Users\DA>getmac  
  
Physical Address      Transport Name  
===== ============  
B4-B6-86-0D-60-71    Media disconnected  
80-C5-F2-7C-3B-59    \Device\Tcpip_{7AF4A79E-A298-41C9-BFB2-27D0A55D94E5}  
80-C5-F2-7C-3B-58    Media disconnected  
0A-00-27-00-00-0C    \Device\Tcpip_{A6724B0B-8BD9-4EE1-AA48-B2E657220F4E}  
C:\Users\DA>
```

3.arp Command

This is used for showing the address resolution cache. This command must be used with a command line switch arp -a is the most common.

```
C:\Users\DA>arp -a  
  
Interface: 192.168.43.31 --- 0xa  
  Internet Address        Physical Address      Type  
  192.168.43.10          54-0e-2d-a5-35-2d  dynamic  
  192.168.43.255         ff-ff-ff-ff-ff-ff  static  
  224.0.0.22              01-00-5e-00-00-16  static  
  224.0.0.251             01-00-5e-00-00-fb  static  
  224.0.0.252             01-00-5e-00-00-fc  static  
  239.255.255.250        01-00-5e-7f-ff-fa  static  
  255.255.255.255        ff-ff-ff-ff-ff-ff  static  
  
Interface: 192.168.56.1 --- 0xc  
  Internet Address        Physical Address      Type  
  192.168.56.255         ff-ff-ff-ff-ff-ff  static  
  224.0.0.22              01-00-5e-00-00-16  static  
  224.0.0.251             01-00-5e-00-00-fb  static  
  224.0.0.252             01-00-5e-00-00-fc  static  
  239.255.255.250        01-00-5e-7f-ff-fa  static  
C:\Users\DA>
```

4. Nbtstat

Diagnostic tool for troubleshooting netBIOS problems.

```
C:\Users\DA>nbtstat -r

NetBIOS Names Resolution and Registration Statistics
-----
Resolved By Broadcast      = 0
Resolved By Name Server    = 0

Registered By Broadcast   = 37
Registered By Name Server = 0

C:\Users\DA>
```

5. Net Command

Used for managing users,service,shares etc..

```
C:\Users\DA>net
The syntax of this command is:

NET
[ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
  HELPMMSG | LOCALGROUP | PAUSE | SESSION | SHARE | START |
  STATISTICS | STOP | TIME | USE | USER | VIEW ] 

C:\Users\DA>
```

Install Apache2

Update your system

sudo apt update

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 259 not upgraded.
vidhya@vidhya-VirtualBox:~/Desktop$ █
```

Install Apache using apt:

sudo apt install apache2

Confirm that Apache is now running with the following command:

sudo systemctl status apache2

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl start apache2
[sudo] password for vidhya:
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres>
   Active: active (running) since Tue 2021-09-28 16:47:25 IST; 3h 56min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 601 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
 Main PID: 769 (apache2)
    Tasks: 6 (limit: 1089)
   Memory: 5.6M
      CPU: 0.000 CPU(s) since start
     CGroup: /system.slice/apache2.service
             └─769 /usr/sbin/apache2 -k start
                 ├─793 /usr/sbin/apache2 -k start
                 ├─794 /usr/sbin/apache2 -k start
                 ├─795 /usr/sbin/apache2 -k start
                 ├─796 /usr/sbin/apache2 -k start
                 └─798 /usr/sbin/apache2 -k start

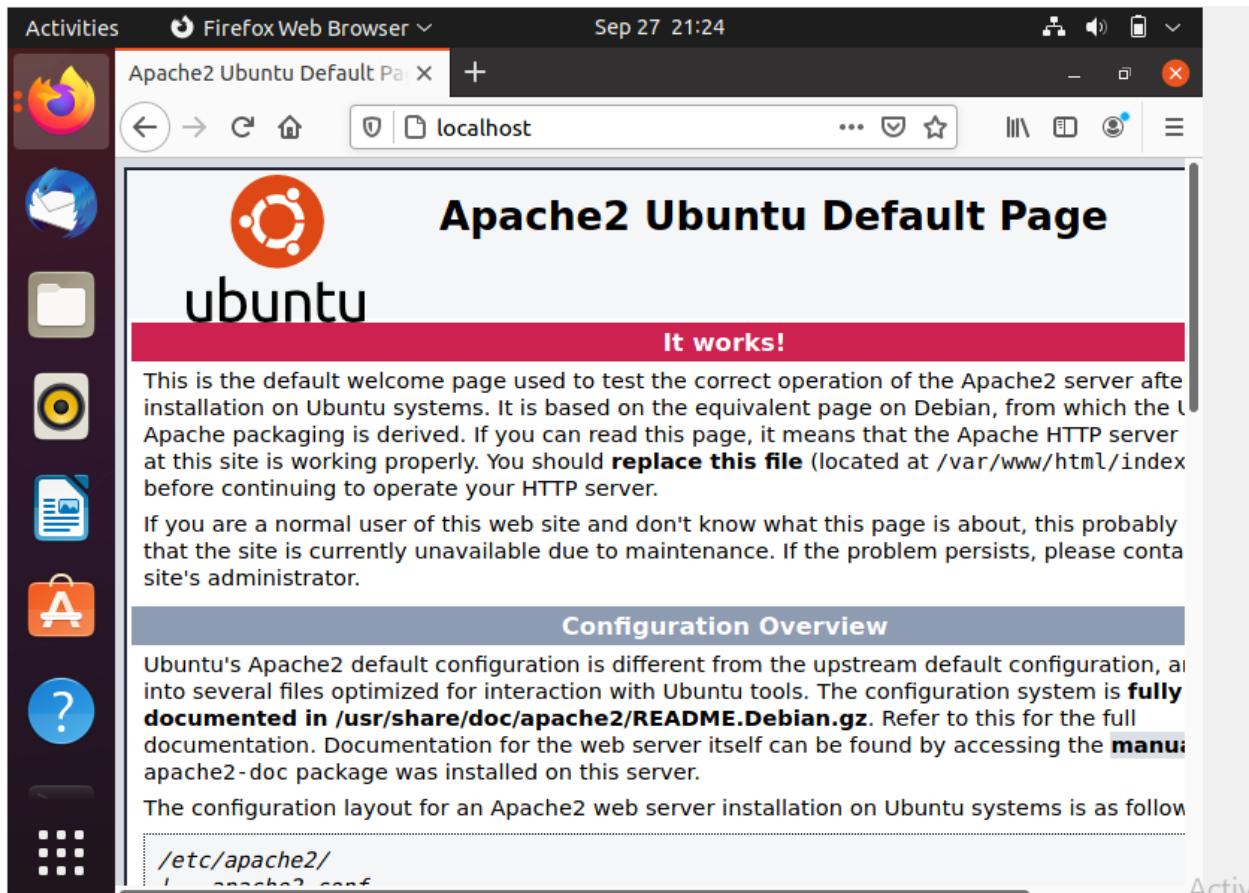
Sep 28 16:47:18 vidhya-VirtualBox systemd[1]: Starting The Apache HTTP Server.>
Sep 28 16:47:25 vidhya-VirtualBox apachectl[628]: AH00558: apache2: Could not >
Sep 28 16:47:25 vidhya-VirtualBox systemd[1]: Started The Apache HTTP Server.

vidhya@vidhya-VirtualBox:~/Desktop$ █
```

Once installed, test by accessing your server's IP in your browser:

<http://127.0.0.1/>

<http://localhost/>



Install mariadb

```
sudo apt install mariadb-server mariadb-client
```

```
sudo systemctl status mysql # to check status
```

```
sudo systemctl start mysql # if not running
```

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl start mysql
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl status mysql
● mariadb.service - MariaDB 10.5.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
   Active: active (running) since Tue 2021-09-28 16:47:30 IST; 3h 59min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 603 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var>
  Process: 619 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_ST>
  Process: 627 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && >
  Process: 857 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_S>
  Process: 859 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/>
 Main PID: 701 (mariadb)
   Status: "Taking your SQL requests now..."
    Tasks: 8 (limit: 1089)
   Memory: 8.8M
      CGroup: /system.slice/mariadb.service
              └─701 /usr/sbin/mariadb

Sep 28 16:47:30 vidhya-VirtualBox mariadb[701]: 2021-09-28 16:47:30 0 [Note] >
Sep 28 16:47:30 vidhya-VirtualBox mariadb[701]: Version: '10.5.12-MariaDB-0ub>
Sep 28 16:47:30 vidhya-VirtualBox systemd[1]: Started MariaDB 10.5.12 database>
Sep 28 16:47:30 vidhya-VirtualBox /etc/mysql/debian-start[862]: Upgrading MySQL>
Sep 28 16:47:31 vidhya-VirtualBox /etc/mysql/debian-start[866]: Looking for 'm>
Sep 28 16:47:31 vidhya-VirtualBox /etc/mysql/debian-start[866]: Looking for 'm>
Sep 28 16:47:31 vidhya-VirtualBox /etc/mysql/debian-start[866]: This installat>
Sep 28 16:47:31 vidhya-VirtualBox /etc/mysql/debian-start[898]: Checking for i>
Sep 28 16:47:31 vidhya-VirtualBox /etc/mysql/debian-start[905]: Triggering myi>
```

Install PHP and commonly used modules

```
sudo apt install php libapache2-mod-php php-ocpache php-cli php-gd php-curl php-mysql
```

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo apt install libapache2-mod-php php-ocpache php-cli php-gd php-curl php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php7.4-ocpache' instead of 'php-ocpache'
libapache2-mod-php is already the newest version (2:7.4+76ubuntu1).
php-cli is already the newest version (2:7.4+76ubuntu1).
php-curl is already the newest version (2:7.4+76ubuntu1).
php-gd is already the newest version (2:7.4+76ubuntu1).
php-mysql is already the newest version (2:7.4+76ubuntu1).
php7.4-ocpache is already the newest version (7.4.16-1ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 259 not upgraded.
vidhya@vidhya-VirtualBox:~/Desktop$
```

```
sudo systemctl restart apache2
```

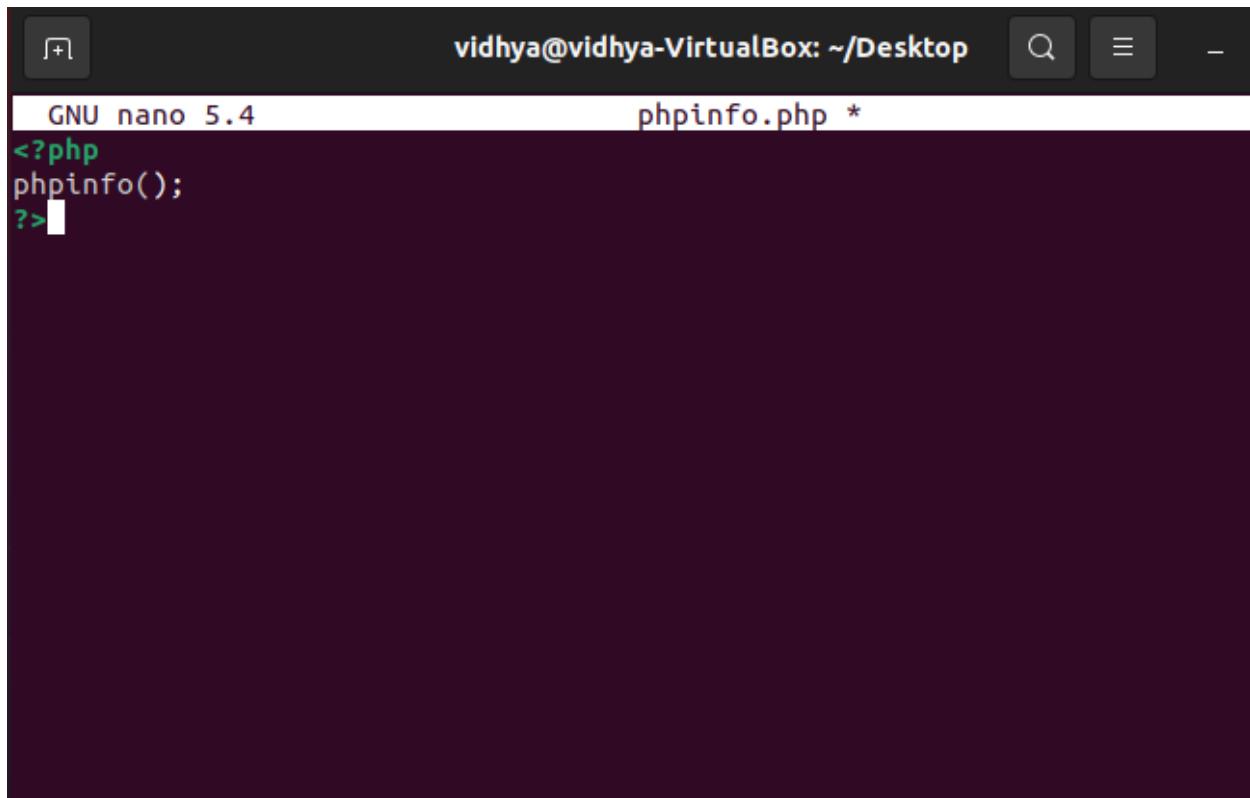
```
[Processing triggers for php7.4-etc (7.4.10-1ubuntu0.1) ...  
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl restart apache2
```

Test PHP Processing on Web Server

```
sudo nano /var/www/html/phpinfo.php
```

Inside the file, type in the valid PHP code:

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



A screenshot of a terminal window titled "vidhya@vidhya-VirtualBox: ~/Desktop". The window shows the command "GNU nano 5.4" followed by the file name "phpinfo.php *". The content of the file is the PHP code: "<?php\n phpinfo();\n?>". The terminal interface includes standard window controls (minimize, maximize, close) and a search bar.

**Press CTRL + X to save and close the file. Press y and ENTER to confirm
Open a browser and type in your IP address/phpinfo.php**

<http://127.0.0.1/phpinfo.php>

PHP Version 7.4.16

System	Linux silja-VirtualBox 5.11.0-34-generic #36-Ubuntu 5.11.0-34-generic
Build Date	Jul 5 2021 13:04:38
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/20-bz2.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-curl.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/20-gd.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/20-iconv.ini, /etc/php/7.4/apache2/20-intl.ini, /etc/php/7.4/apache2/20-mbstring.ini, /etc/php/7.4/apache2/20-mysqli.ini, /etc/php/7.4/apache2/20-pcre.ini, /etc/php/7.4/apache2/20-pdo_mysql.ini, /etc/php/7.4/apache2/20-pdo_oci.ini, /etc/php/7.4/apache2/20-pdo_pgsql.ini, /etc/php/7.4/apache2/20-pdo_sqlite.ini, /etc/php/7.4/apache2/20-readline.ini, /etc/php/7.4/apache2/20-sqlite3.ini, /etc/php/7.4/apache2/20-zip.ini, /etc/php/7.4/apache2/conf.d/20-xdebug.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini, /etc/php/7.4/apache2/conf.d/20-zipArchive.ini, /etc/php/7.4/apache2/conf.d/20-zipArchiveGZ.ini

Install phpmyadmin

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo apt install phpmyadmin php-mbstring ph  
p-zip php-zip php-gd php-json php-curl  
[sudo] password for vidhya:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
php-curl is already the newest version (2:7.4+76ubuntu1).  
php-gd is already the newest version (2:7.4+76ubuntu1).  
The following additional packages will be installed:  
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common  
  libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5  
  LibreOffice Writer libcon-qr-code php-bz2 php-dasprid-enum php-google-recaptcha  
  libphpmyadmin-motranslator php-phpmyadmin-shapefile  
  php-phpmyadmin-sql-parser php-phpseclib php-psr-cache php-psr-container  
  php-psr-log php-symfony-cache php-symfony-cache-contracts  
  php-symfony-expression-language php-symfony-service-contracts  
  php-symfony-var-exporter php-tcpdf php-twig php-twig-extensions php-xml  
  php7.4-bz2 php7.4-mbstring php7.4-xm php7.4-zip  
Suggested packages:  
  php-imagick php-dbase php-libodium php-mcrypt php-gmp  
  php-symfony-service-implementation php-twig-doc php-symfony-translation  
  php-recode php-gd2 php-pragmarx-google2fa php-samyoul-u2f-php-server  
Recommended packages:  
  php-mcrypt  
The following NEW packages will be installed:  
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common  
  libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5  
  libzip4 php-bacon-qr-code php-bz2 php-dasprid-enum php-google-recaptcha  
  php-json php-mbstring php-phpmyadmin-motranslator php-phpmyadmin-shapefile
```

```
sudo systemctl restart apache2
```

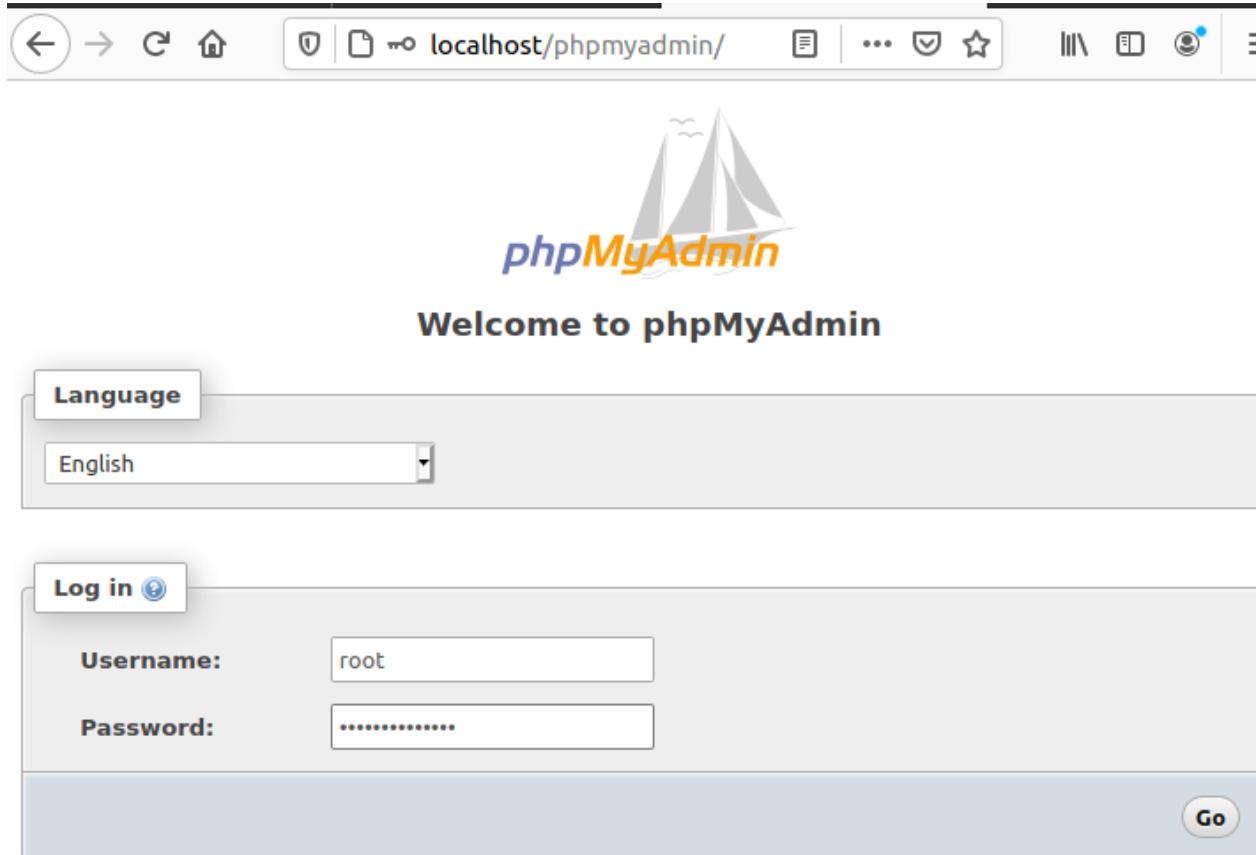
```
Processing triggers for php7.4-clients (7.4.10-1ubuntu0.2.1) ...  
vidhya@vidhya-VirtualBox:~/Desktop$ sudo systemctl restart apache2
```

Open a browser : http://localhost/phpmyadmin

username : root

password : yourpasswordIf phpmyadmin page not found :

`nano /etc/apache2/apache2.conf`



Add this line to last of the file.

Press CTRL + X to save and close the file. Press y and ENTER to confirm

`Include /etc/phpmyadmin/apache.conf`

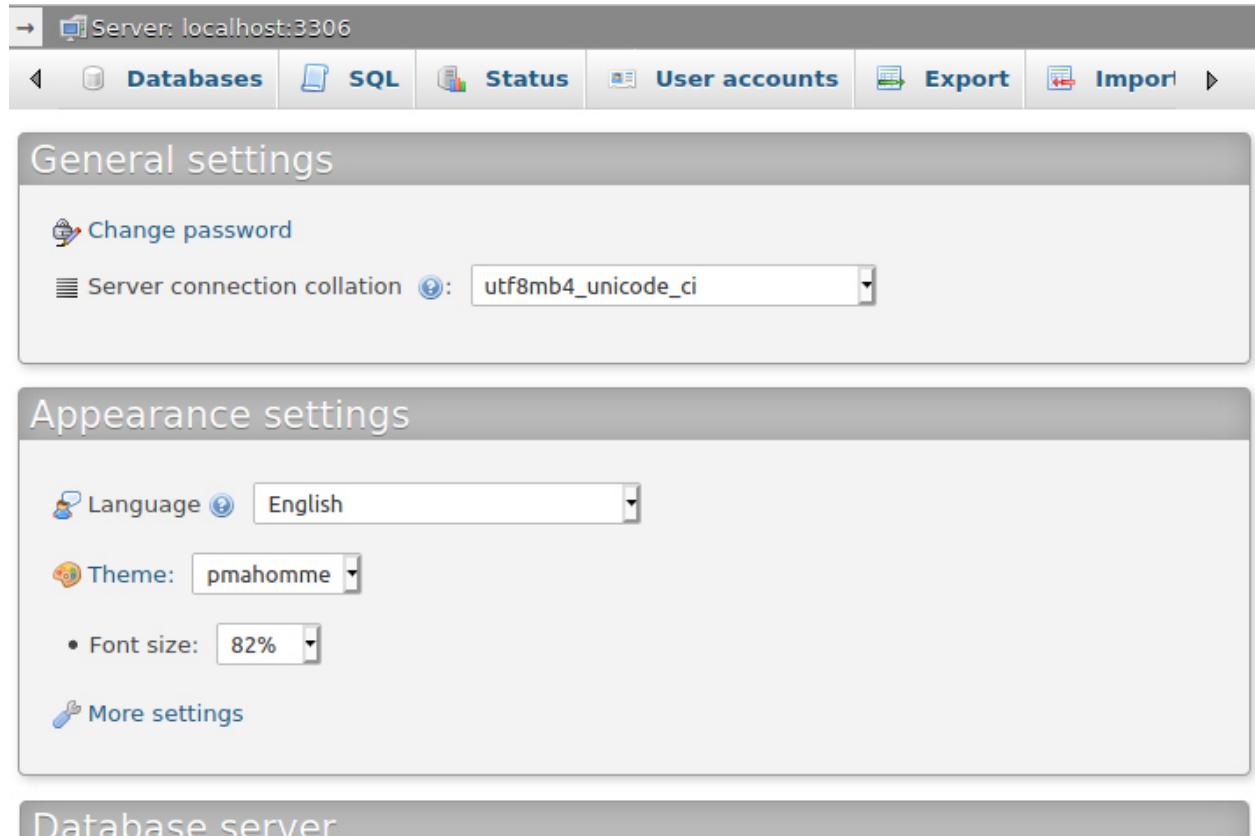
restart apache2 - now try : http://localhost/phpmyadmin

```
sudo systemctl restart apache2
```

If any problem for login run the following command

```
sudo mysql
```

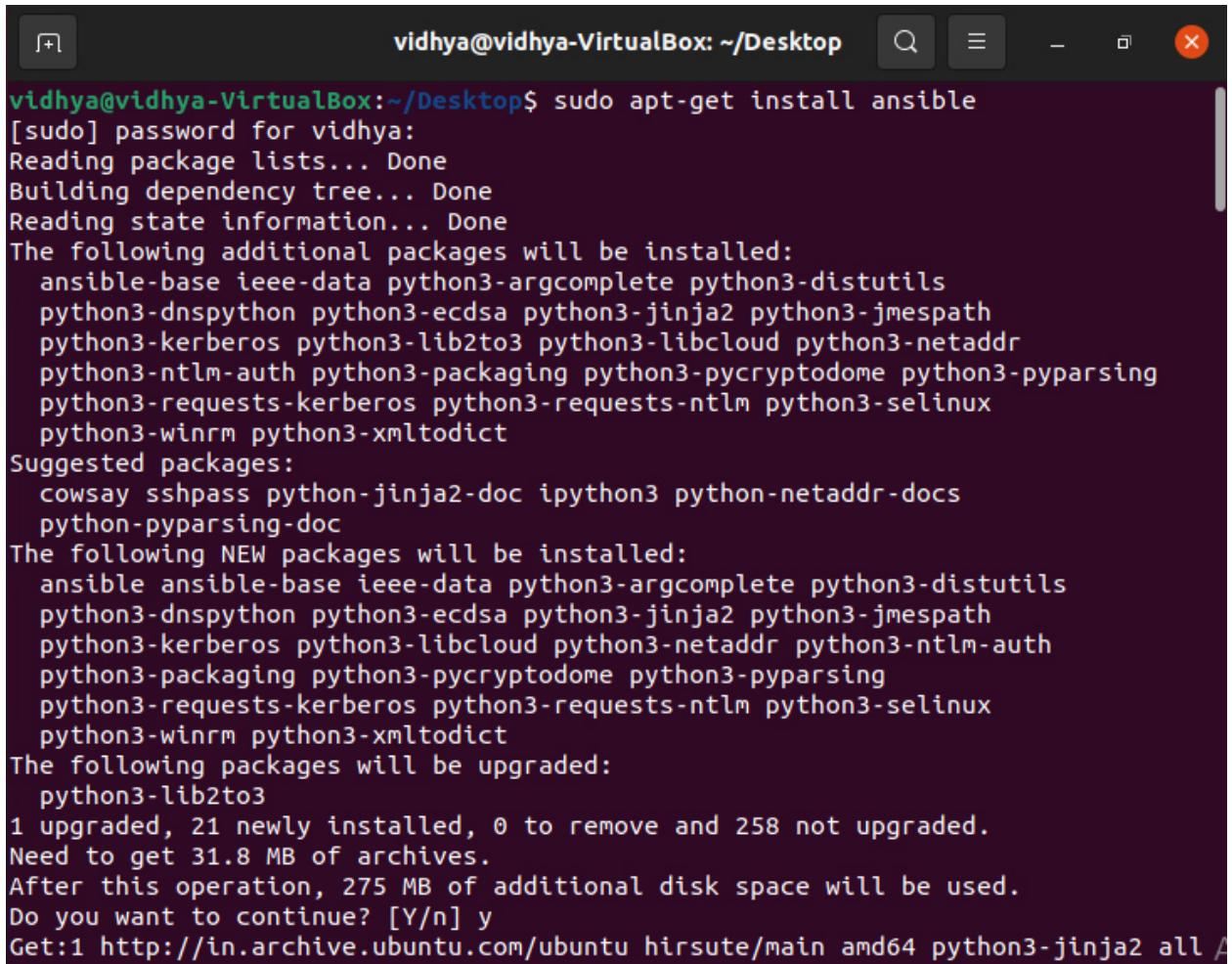
```
ALTER USER root@localhost IDENTIFIED BY "yourpassword";
```



Ansible Installation:

Syntax:

```
$ sudo apt-get install ansible
```

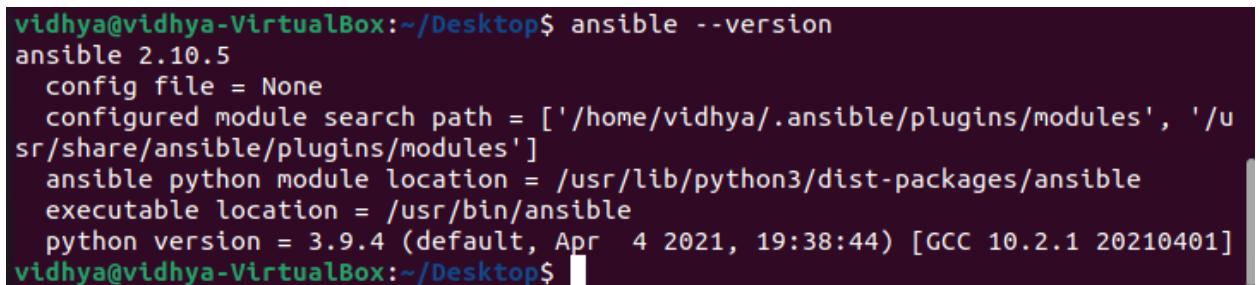


The screenshot shows a terminal window titled "vidhya@vidhya-VirtualBox: ~/Desktop". The terminal displays the output of the command \$ sudo apt-get install ansible. It shows the password entry, package lists, dependency tree, state information, additional packages to be installed (including ansible-base, python3-argcomplete, python3-distutils, etc.), suggested packages (cowsay, sshpass, etc.), new packages to be installed (ansible, ansible-base, etc.), packages to be upgraded (python3-lib2to3), and the upgrade summary (1 upgraded, 21 newly installed, 0 to remove and 258 not upgraded). It also shows the need for 31.8 MB of disk space and asks if the user wants to continue.

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo apt-get install ansible
[sudo] password for vidhya:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-base ieee-data python3-argcomplete python3-distutils
  python3-dnspython python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-lib2to3 python3-libcloud python3-netaddr
  python3-ntlm-auth python3-packaging python3-pycryptodome python3-pyparsing
  python3-requests-kerberos python3-requests-ntlm python3-selinux
  python3-winrm python3-xmldict
Suggested packages:
  cowsay sshpass python-jinja2-doc ipython3 python-netaddr-docs
  python-pyparsing-doc
The following NEW packages will be installed:
  ansible ansible-base ieee-data python3-argcomplete python3-distutils
  python3-dnspython python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-packaging python3-pycryptodome python3-pyparsing
  python3-requests-kerberos python3-requests-ntlm python3-selinux
  python3-winrm python3-xmldict
The following packages will be upgraded:
  python3-lib2to3
1 upgraded, 21 newly installed, 0 to remove and 258 not upgraded.
Need to get 31.8 MB of archives.
After this operation, 275 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://in.archive.ubuntu.com/ubuntu hirsute/main amd64 python3-jinja2 all A
```

Syntax:

```
$ ansible --version
```



The screenshot shows a terminal window titled "vidhya@vidhya-VirtualBox: ~/Desktop". The terminal displays the output of the command \$ ansible --version. It shows the version (2.10.5), config file (None), configured module search path ('/home/vidhya/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'), ansible python module location (/usr/lib/python3/dist-packages/ansible), executable location (/usr/bin/ansible), and python version (3.9.4).

```
vidhya@vidhya-VirtualBox:~/Desktop$ ansible --version
ansible 2.10.5
  config file = None
  configured module search path = ['/home/vidhya/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.9.4 (default, Apr  4 2021, 19:38:44) [GCC 10.2.1 20210401]
vidhya@vidhya-VirtualBox:~/Desktop$
```

TCPDUMP

Tcpdump Installation

On Debian based distributions tcpdump can be installed with the APT command :

```
#sudo apt update && sudo apt install tcpdump
```

```
vidhya@vidhya-VirtualBox:~/Desktop$ sudo apt update && sudo apt install tcpdump
[sudo] password for vidhya:
Hit:1 http://in.archive.ubuntu.com/ubuntu hirsute InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu hirsute-updates InRelease [115 kB]
Get:3 http://security.ubuntu.com/ubuntu hirsute-security InRelease [110 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu hirsute-backports InRelease [101 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu hirsute-updates/main amd64 Packages [387 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu hirsute-updates/main i386 Packages [195 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu hirsute-updates/main Translation-en [102 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu hirsute-updates/main amd64 DEP-11 Metadata [95.0 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu hirsute-updates/main amd64 c-n-f Metadata [7,440 B]
Get:10 http://in.archive.ubuntu.com/ubuntu hirsute-updates/universe amd64 Packages [322 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu hirsute-updates/universe i386 Packages [234 kB]
Get:12 http://security.ubuntu.com/ubuntu hirsute-security/main amd64 DEP-11 Metadata [9,716 B]
Get:13 http://in.archive.ubuntu.com/ubuntu hirsute-updates/universe amd64 DEP-1 Metadata [56.7 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu hirsute-updates/universe amd64 c-n-f Metadata [7,012 B]
Get:15 http://in.archive.ubuntu.com/ubuntu hirsute-updates/multiverse amd64 DEP-11 Metadata [944 B]
```

To invoke tcpdump without any options and filters:

- sudo tcpdump

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump
[sudo] password for vidhya:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
20:13:44.749965 IP vidhya-VirtualBox.42049 > golem.canonical.com.ntp: NTPv4, client, length 48
20:13:44.786669 IP vidhya-VirtualBox.39677 > 192.168.43.123.domain: 39341+ PTR? 199.89.189.91.in-addr.arpa. (44)
20:13:45.243335 IP golem.canonical.com.ntp > vidhya-VirtualBox.42049: NTPv4, server, length 48
20:13:46.024553 IP 192.168.43.123.domain > vidhya-VirtualBox.39677: 39341 1/0/0 PTR golem.canonical.com. (77)
20:13:46.027293 IP vidhya-VirtualBox.48254 > 192.168.43.123.domain: 56560+ PTR? 15.2.0.10.in-addr.arpa. (40)
20:13:46.329791 IP 192.168.43.123.domain > vidhya-VirtualBox.48254: 56560 NXDomain* 0/1/0 (99)
20:13:46.336788 IP vidhya-VirtualBox.45400 > 192.168.43.123.domain: 55946+ PTR? 123.43.168.192.in-addr.arpa. (45)
20:13:46.534364 IP 192.168.43.123.domain > vidhya-VirtualBox.45400: 55946 NXDomain* 0/1/0 (104)
20:13:49.937544 ARP, Request who-has _gateway tell vidhya-VirtualBox, length 28
20:13:49.939173 ARP, Reply _gateway is-at 52:54:00:12:35:02 (oui Unknown), length 46
20:13:49.940307 IP vidhya-VirtualBox.45021 > 192.168.43.123.domain: 52469+ PTR? 2.2.0.10.in-addr.arpa. (39)
20:13:50.010676 IP 192.168.43.123.domain > vidhya-VirtualBox.45021: 52469 NXDomain* 0/1/0 (98)
20:15:53.249389 IP vidhya-VirtualBox.49784 > golem.canonical.com.ntp: NTPv4, client, length 48
```

```
$ sudo tcpdump -i any
```

```
vidhya@vidhya-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
20:18:59.565240 IP localhost.45963 > localhost.domain: 2624+ [1au] A? api.snapc
raft.io. (45)
20:18:59.568936 IP localhost.36240 > localhost.domain: 60629+ [1au] PTR? 53.0.0
.127.in-addr.arpa. (52)
20:18:59.570211 IP vidhya-VirtualBox.44462 > 192.168.43.123.domain: 5521+ A? ap
i.snapcraft.io. (34)
20:18:59.571049 IP localhost.domain > localhost.36240: 60629 1/0/1 PTR localhos
t. (75)
20:18:59.571911 IP localhost.45963 > localhost.domain: 6067+ [1au] AAAA? api.sn
apcraft.io. (45)
20:18:59.575986 IP localhost.49945 > localhost.domain: 1883+ [1au] PTR? 123.43.
168.192.in-addr.arpa. (56)
20:18:59.576816 IP vidhya-VirtualBox.40080 > 192.168.43.123.domain: 34059+ PTR?
123.43.168.192.in-addr.arpa. (45)
20:18:59.852336 IP vidhya-VirtualBox.40686 > api.snapcraft.io.https: Flags [S],
seq 1781498383, win 64240, options [mss 1460,sackOK,TS val 3859968223 ecr 0,no
p,wscale 7], length 0
20:18:59.853609 IP localhost.49659 > localhost.domain: 41071+ [1au] PTR? 38.92.
189.91.in-addr.arpa. (54)
20:18:59.869780 IP vidhya-VirtualBox.36239 > 192.168.43.123.domain: 37707+ PTR?
38.92.189.91.in-addr.arpa. (43)
20:19:00.230266 IP api.snapcraft.io.https > vidhya-VirtualBox.40686: Flags [S.]
, seq 3392001, ack 1781498384, win 65535, options [mss 1460], length 0
20:19:00.230351 TP vidhya-VirtualBox.40686 > api.snapcraft.io.https: Flags [F.]
A
```

```
# tcpdump -D
```

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

To capture packets flowing through a specific interface, use the `-i` flag with the interface name. Without the `-i` interface tcpdump will pick up the first network interface it comes across.

```
# tcpdump -i enp2s0
```

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
20:24:16.121866 IP vidhya-VirtualBox.59228 > bahamutt.canonical.com.https: Flag
s [..], ack 24372650, win 65535, length 0
20:24:16.123430 IP vidhya-VirtualBox.34765 > 192.168.43.123.domain: 59070+ PTR?
  Files 0.10.in-addr.arpa. (40)
20:24:16.131253 IP 192.168.43.123.domain > vidhya-VirtualBox.34765: 59070 NXDom
ain 0/0/0 (40)
20:24:16.132086 IP vidhya-VirtualBox.35487 > 192.168.43.123.domain: 34935+ PTR?
  123.43.168.192.in-addr.arpa. (45)
20:24:16.132722 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 1:1361, ack 0, win 65535, length 1360
20:24:16.135926 IP 192.168.43.123.domain > vidhya-VirtualBox.35487: 34935 NXDom
ain 0/0/0 (45)
20:24:16.145353 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 1361:2721, ack 0, win 65535, length 1360
20:24:16.145389 IP vidhya-VirtualBox.59228 > bahamutt.canonical.com.https: Flag
s [..], ack 2721, win 65535, length 0
20:24:16.150608 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 2721:4081, ack 0, win 65535, length 1360
20:24:16.203104 IP vidhya-VirtualBox.59228 > bahamutt.canonical.com.https: Flag
s [..], ack 4081, win 65535, length 0
20:24:16.219600 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 4081:6801, ack 0, win 65535, length 2720
20:24:16.219636 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 6801:8161, ack 0, win 65535, length 1360
20:24:16.219782 IP vidhya-VirtualBox.59228 > bahamutt.canonical.com.https: Flag
s [..], ack 8161, win 65535, length 0
```

To capture only a set of lines, say 5, use the `-c` flag:

```
#tcpdump -c 5
```

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
20:25:12.592220 IP bahamutt.canonical.com.https > vidhya-VirtualBox.59228: Flag
s [P..], seq 28230970:28232330, ack 4118352563, win 65535, length 1360
20:25:12.603159 IP vidhya-VirtualBox.47549 > 192.168.43.123.domain: 12535+ PTR?
  15.2.0.10.in-addr.arpa. (40)
20:25:12.615134 IP 192.168.43.123.domain > vidhya-VirtualBox.47549: 12535 NXDom
ain 0/0/0 (40)
20:25:12.617526 IP vidhya-VirtualBox.47805 > 192.168.43.123.domain: 30071+ PTR?
  123.43.168.192.in-addr.arpa. (45)
20:25:12.623662 IP 192.168.43.123.domain > vidhya-VirtualBox.47805: 30071 NXDom
ain 0/0/0 (45)
5 packets captured
7 packets received by filter
0 packets dropped by kernel
vidhya@vidhya-VirtualBox:~$
```

port filter

Use port filter to view packets arriving at a specific port:

```
#sudo tcpdump -i enp2so -c 5 port 80
```

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump -i enp0s3 -c 5 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
vidhya@vidhya-VirtualBox:~$
```

host filter

To capture all packets arriving at or leaving from the host with IP address of 10.0.2.15:

```
# tcpdump host 10.0.2.15
```

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump host 10.0.2.15
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
20:34:22.749371 IP vidhya-VirtualBox.38154 > golem.canonical.com.ntp: NTPv4, cl
ient, length 48
20:34:22.860386 IP vidhya-VirtualBox.41222 > 192.168.43.123.domain: 61227+ PTR?
15.2.0.10.in-addr.arpa. (40)
20:34:22.934477 IP 192.168.43.123.domain > vidhya-VirtualBox.41222: 61227 NXDom
ain 0/0/0 (40)
20:34:22.955480 IP vidhya-VirtualBox.36229 > 192.168.43.123.domain: 58720+ PTR?
123.43.168.192.in-addr.arpa. (45)
20:34:23.035470 IP 192.168.43.123.domain > vidhya-VirtualBox.36229: 58720 NXDom
ain 0/0/0 (45)
20:34:23.240186 IP golem.canonical.com.ntp > vidhya-VirtualBox.38154: NTPv4, Se
rver, length 48
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
vidhya@vidhya-VirtualBox:~$
```

To capture packets of a specific protocol type, for example, icmp, on eth1 interface:

```
# tcpdump -i eth1 icmp  
#sudo tcpdump -n net 10.1
```

```
vidhya@vidhya-VirtualBox:~$ sudo tcpdump -n net 10.10  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes  
^C  
0 packets captured  
0 packets received by filter  
0 packets dropped by kernel  
vidhya@vidhya-VirtualBox:~$
```

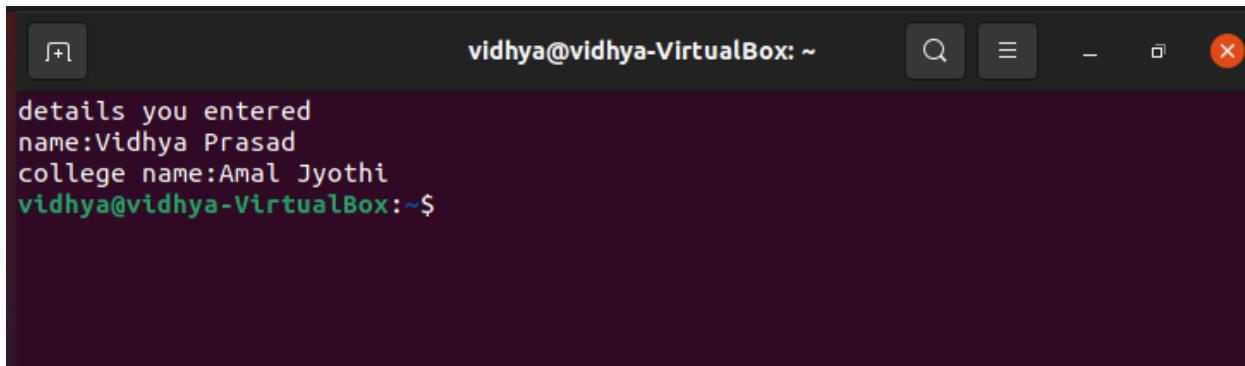
Saving packet headers to a file

- ◎ Since the output of tcpdump can scroll past the screen quite fast, you can store packet headers to a file with the -w flag. The files to save the output use pcap format and have an extension of .pcap.
- ◎ PCAP stands for packet capture. The following command saves 10 lines of output on the eth1 interface to icmp.pcap.
- ◎ # tcpdump -i eth1 -c 10 -w icmp.pcap
- ◎ You can read this file with -r flag:
- ◎ tcpdump -r icmp.pcap Viewing packet details
- ◎ So far we have only seen the packet headers, to view packet contents use -A option. This prints the packet contents in ASCII, which can be of help in network troubleshooting. Also -X flag can be used to display output in hex format. This may not be of much help if the connection is encrypted.

```
# tcpdump -c10 -i eth1 -n -A port 80
```

1. Write a shell script to ask your name, and college name and print it on the screen.

```
echo "enter details and view"
echo enter your name
read name
echo enter your college name read c
clear
echo Details you entered
echo Name:$name
echo College:$c
```



```
details you entered
name:Vidhya Prasad
college name:Amal Jyothi
vidhya@vidhya-VirtualBox:~$
```

2. Write a shell script to set a value for a variable and display it on command line interface.

```
echo "Display value of a variable" a=10
echo $a
vidhya@vidhya-VirtualBox:~$ vi bash2.sh
vidhya@vidhya-VirtualBox:~$ ./bash2.sh
Show value of the value
a=10

vidhya@vidhya-VirtualBox:~$
```

3. Write a shell script to perform addition, subtraction, multiplication, division with two numbers that is accepted from user.

```
echo enter a number read
a
echo enter another number
read b
echo enter operation
echo "\n1.addition \n2.subtraction \n3.multiplication \n4.division"
```

```
read op
case "$op" in
"1") echo "a+b=$((a+b));;
"2") echo "a-b=$((a-$b));;
"3") echo "a*b=$((a*$b));;
"4") echo "a/b=$((a/$b));;
esac
```

```
user@user-VirtualBox:~$ bash 3.sh
enter a number
7
enter another number
8
enter operation
\n1.addition \n2.subtraction \n3.multiplication \n4.division
2
a-b=-1
```

4. Write a shell script to check the value of a given number and display whether the number is found or not.

```
echo enter a number read
a
if [ $a -eq 10 ];
then
echo "number found"
else
echo "not found"
fi
```

```
user@user-VirtualBox:~$ bash 4.sh
enter a number
9
not found
```

5. Write a shell script to display current date, calendar.

```
echo "Today is $(date)"
echo "calender:"
cal
```

```
user@user-VirtualBox:~$ bash 5.sh
Today is Saturday 02 October 2021 05:53:45 PM IST
calender:
    October 2021
Su Mo Tu We Th Fr Sa
        1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

6. Write a shell script to check a number is even or odd.

```
#!/bin/bash
```

```
echo enter a number
```

```
read n
```

```
x=$(( $n % 2 ))
```

```
if [ $x -eq 0 ];
```

```
then
```

```
echo "number is even"
```

```
else
```

```
echo "number is odd"
```

```
fi
```

```
user@user-VirtualBox:~$ bash 6.sh
enter a number
4
number is even
```

7. Write a shell script to check a number is greater than, less than or equal to another number.

```
echo enter first number
read a
echo enter second number
read b
if [ $a -gt $b ];
then
echo "$a is larger"
elif [ $b -gt $a ];
then
echo "$b is larger"
else
echo "both are equal"
fi
```

```
user@user-VirtualBox:~$ bash 7.sh
enter first number
54
enter second number
34
54 is larger
```

8. Write a shell script to find the sum of first 10 numbers.

```
s=0
for ((i=0;i<=10;i++)) do
s=`expr $s + $i`
done
echo "sum of first 10 numbers=$s"
```

```
user@user-VirtualBox:~$ bash 8.sh
sum of first 10 numbers=55
```

9. Write a shell script to find the sum, the average and the product of the four integers entered.

```
echo please enter your first number
read a
echo please enter your second number
read b
echo please enter your third number
read c
echo please enter your fourth number
read d
sum=$((a + b + c + d))
prod=$((a * b * c * d))
avg=$(echo $sum/4 | bc -l)
echo "the sum is:$sum echo
"the average is:$avg echo
"the product is:$prod
```

```
user@user-VirtualBox:~$ bash 9.sh
please enter your first number
1
please enter your second number
2
please enter your third number
3
please enter your fourth number
4
the sum is:10
the average is:2.5000000000000000000000000000000
the product is:24
```

10. Write a shell script to find the smallest of three numbers.

```
echo enter first number
read a
echo enter second number
read b
echo enter third number
read c
if [ $a -lt $b ];
then
if [ $a -lt $c ];
then
echo "$a is smallest"
fi
elif [ $b -lt $c ];
then
echo "$b is smallest"
else
echo "$c is smallest";
fi
```

```
user@user-VirtualBox:~$ bash 10.sh
enter first number
5
enter second number
2
enter third number
6
2 is smallest
```

11. Write a shell program to find factorial of given number.

```
echo enter a number read n
f=1
for ((i=2;i<=n;i++))
do
f=$((f*i))
done
echo "factorial is $f"
```

```
user@user-VirtualBox:~$ hash 11.sh
enter a number
5
factorial is 120
```

12. Write a shell program to check a number is palindrome or not.

```
echo enter a number read n
rev=$(echo $n | rev)
if [ $n -eq $rev ];
then
echo "number is palindrome"
else
echo "number is not palindrome"
fi
```

```
user@user-VirtualBox:~$ bash 12.sh
enter a number
1221
number is palindrome
```

13. Write a shell script to find the average of the numbers entered in command line.

```
echo enter size read n
i=1 s=0
echo "enter numbers"
while [ $i -le $n ]
do
read num s=$((s+num))
i=$((i+1))
done
avg=$(echo $s/$n | bc -l)
echo "average is $avg"
```

```
user@user-VirtualBox:~$ bash 13.sh
enter size
5
enter numbers
6
7
8
9
4
average is 6.80000000000000000000000000000000
```

14. Write a shell program to find the sum of all the digits in a number.

echo enter a number read

```
n  
s=0  
while [ $n -gt 0 ]  
do  
mod=$((n%10))  
s=$((s+mod))  
n=$((n/10))  
done  
echo "sum of digit is $s"
```

```
user@user-VirtualBox:~$ bash 14.sh  
enter a number  
678  
sum of digit is 21
```

15. Write a shell Script to check whether given year is leap year or not.

```
echo enter year read  
y a=$((y%4))  
b=$((y%100))  
c=$((y%400))  
if [ $a -eq 0 -a $b -ne 0 -o $c -eq 0 ];  
then  
echo "$y is leap year"  
else  
echo "$y is leap year"  
fi
```

```
user@user-VirtualBox:~$ bash 15.sh  
enter year  
1994  
1994 is leap year
```

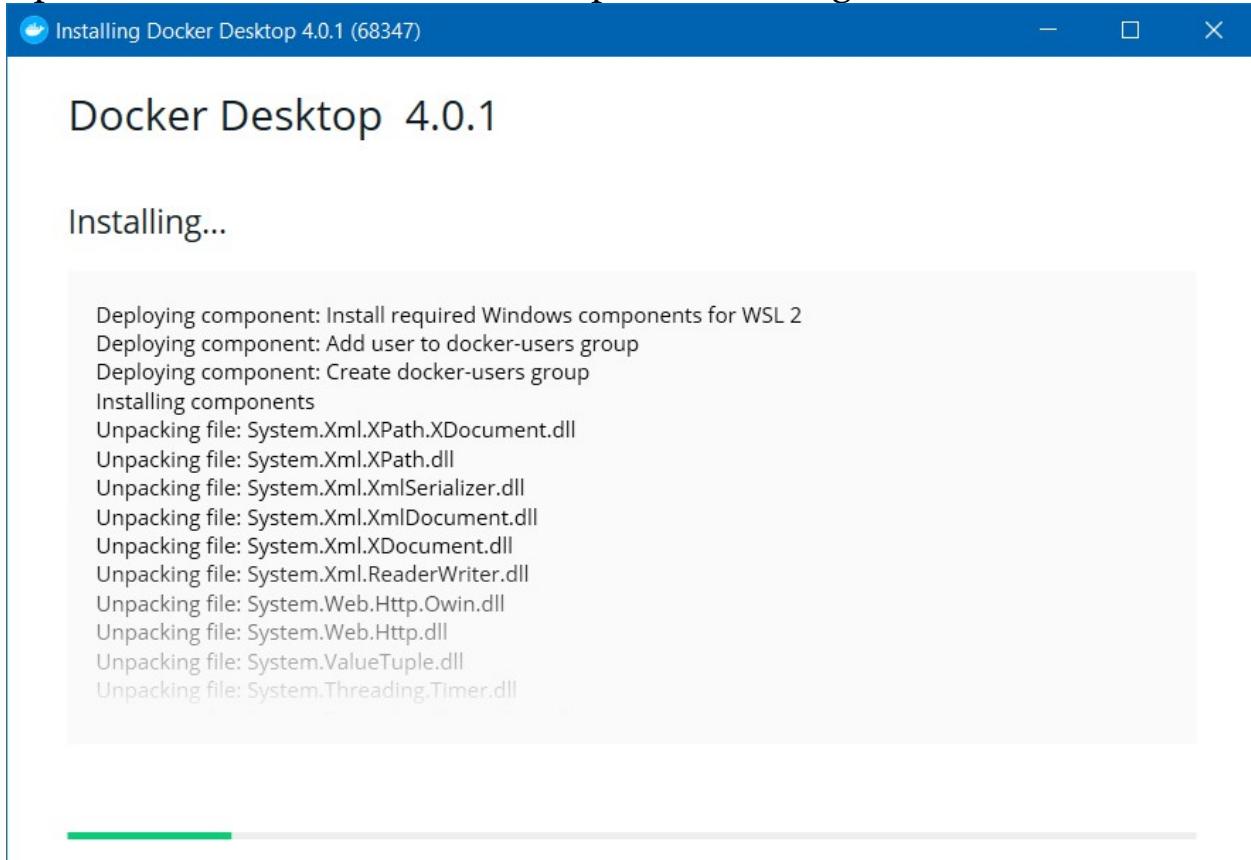
Step-I

Download Docker Desktop installer for Windows from
<https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe>



Step-II

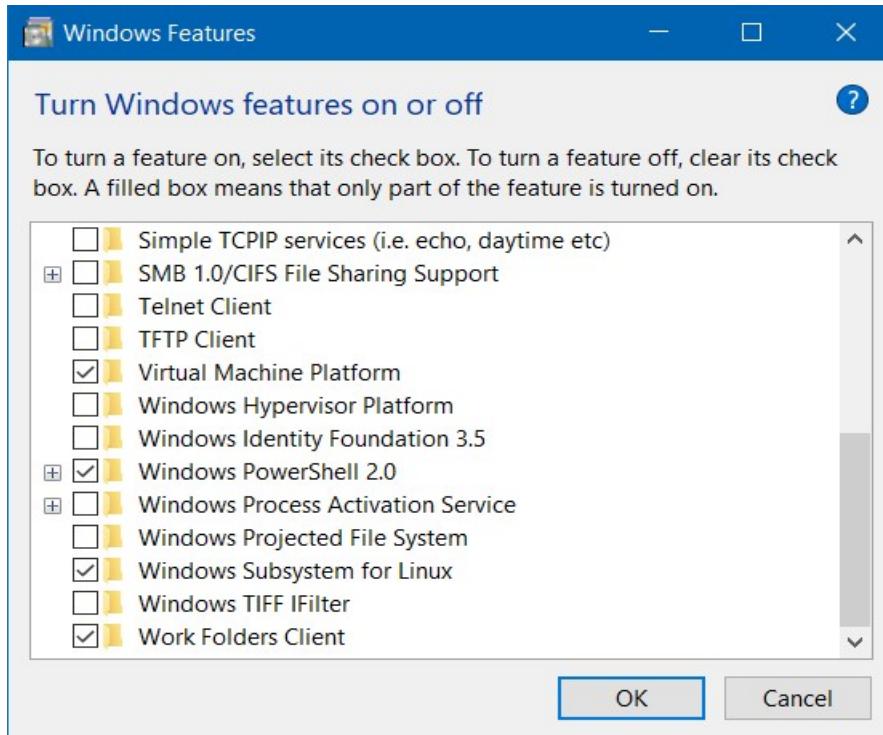
Open the .exe file and follow the steps after clicking install button.



Step-III

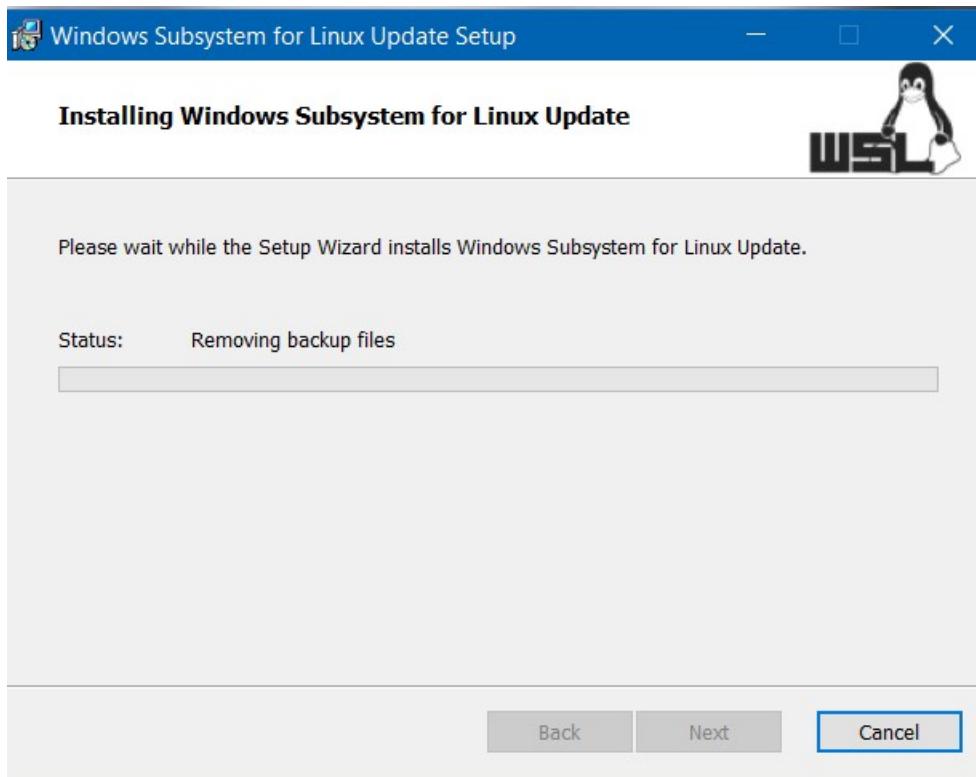
Once installed go to programs and features and click turn on windows features on or off

Scroll to the bottom and select windows subsystem for Linux



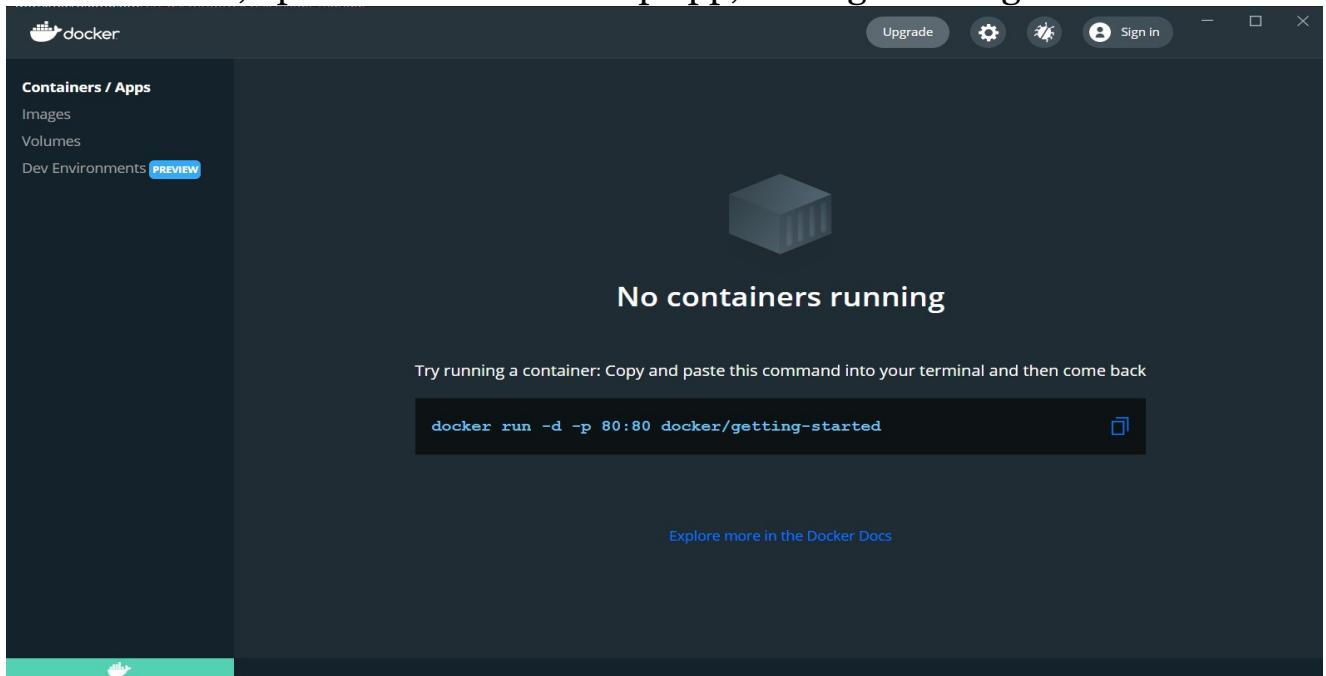
Step-IV

If any WSL 2 error occurs download windows subsystem for linux update package and install the .exe file, after the installation restart the windows device.



Step-V

Once installed, open the docker desktop app, and signin using the dockerID



Step-VI

Now pull any image from docker hub using the docker pull command in the command prompt (eg: docker pull ubuntu)

A screenshot of a Windows Command Prompt window titled 'Administrator: Command Prompt'. The title bar is blue with white text. The window shows a black background with white text. The user has run the command 'docker run -d -p 80:80 docker/getting-started'. The output shows an error message: 'Unable to find image 'docker/getting-started:latest' locally'. It then attempts to connect to Docker Hub: 'docker: Error response from daemon: Get "https://registry-1.docker.io/v2/": dial tcp: lookup registry-1.docker.io on 192.168.65.5:53: no such host.' followed by a help message: 'See 'docker run --help''. The user then runs 'docker pull ubuntu'. The output shows the image being pulled: 'Using default tag: latest', 'latest: Pulling from library/ubuntu', 'f3ef4ff62e0d: Pull complete', 'Digest: sha256:65de08a8dabf289ef114053ab32f79e0c333a4fbfa1fe3778bb13ae921a7849b', 'Status: Downloaded newer image for ubuntu:latest', and 'docker.io/library/ubuntu:latest'. The command prompt ends with 'C:\Windows\system32>'.

Now in the images tab an image of ubuntu will be displayed, we can run the ubuntu instance using the cli.

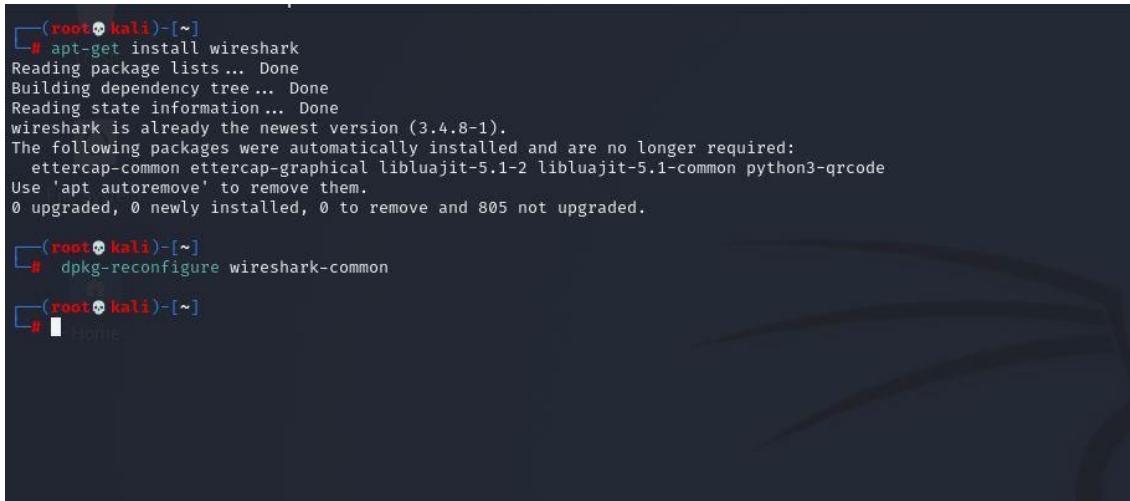
The screenshot shows the Docker interface for managing images. On the left, there's a sidebar with options: Containers / Apps, Images (which is selected and highlighted in blue), Volumes, and Dev Environments (with a 'PREVIEW' button). The main area is titled 'Images on disk' and shows '1 images' with a total size of '72.78 MB'. There are two tabs at the top: 'LOCAL' (which is active) and 'REMOTE REPOSITORIES'. Below is a search bar and a checkbox for 'In Use only'. A table lists the image details:

NAME	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	597ce1600cf4	about 8 hours ago	72.78 MB

Analyzing network packet stream using nc and wireshark

Step1:

- **sudo apt-get install wireshark**



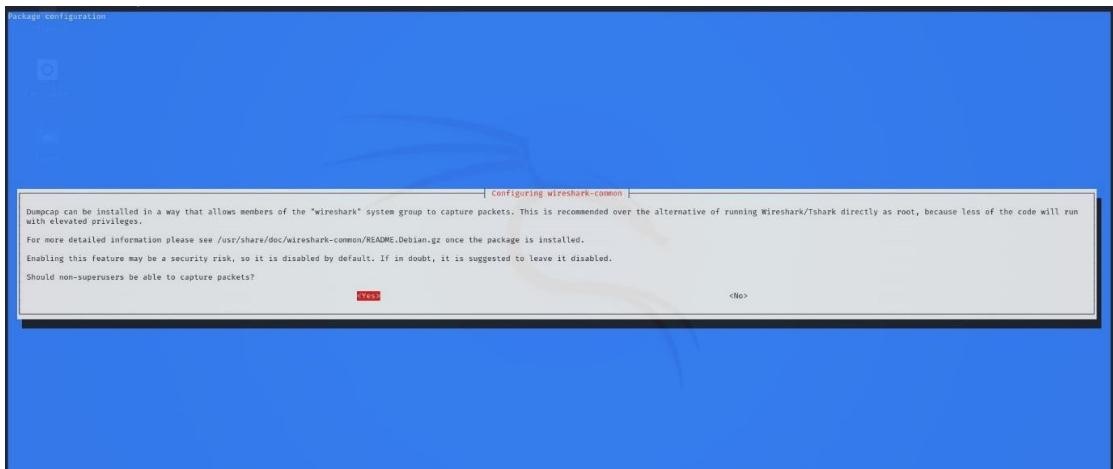
```
[root@kali] ~
# apt-get install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wireshark is already the newest version (3.4.8-1).
The following packages were automatically installed and are no longer required:
  ettercap-common ettercap-graphical libluajit-5.1-2 libluajit-5.1-common python3-qrcode
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 805 not upgraded.

[root@kali] ~
# dpkg-reconfigure wireshark-common

[root@kali] ~
# [Home]
```

Step2:

- **sudo dpkg-reconfigure wireshark-common**



Step3.

open wireshark from the applist

