**Linux**

is an open source multi-tasking, multi-user operating system.

It was initially developed by Linus Torvalds in 1991. Linux OS is widely used in desktops,

mobiles, mainframes etc.

**Unix**

Unix is multitasking, multi-user operating system but is not free to use and is not open source.

It was developed in 1969 by Ken Thompson team at AT&T Bell Labs. It is widely used on servers,

workstations etc. Following are the important differences between Linux and Unix.

|  |  |  |
| --- | --- | --- |
| **Comparison** | **Linux** | **Unix** |
| Definition | It is an open-source operating system which is *freely available to everyone*. | It is an operating system which *can be only used by its copyrighters*. |
| Examples | It has different distros like Ubuntu, Redhat, Fedora, etc | IBM AIX, HP-UX and Sun Solaris. |
| Users | Nowadays, Linux is in great demand. Anyone can use Linux whether a home user, developer or a student. | It was developed mainly for servers, workstations and mainframes. |
| Usage | Linux is used everywhere from servers, PC, smartphones, tablets to mainframes and supercomputers. | It is used in servers, workstations and PCs. |
| Cost | Linux is freely distributed,downloaded, and distributed through magazines also. And priced distros of Linux are also cheaper than Windows. | Unix copyright vendors decide different costs for their respective Unix Operating systems. |
| Development | As it is open source, it is developed by sharing and collaboration of codes by world-wide developers. | Unix was developed by AT&T Labs, various commercial vendors and non-profit organizations. |
| Manufacturer | Linux kernel is developed by the community of developers from different parts of the world. Although the father of Linux, Linus Torvalds oversees things. | Unix has three distributions IBM AIX, HP-UX and Sun Solaris. Apple also uses Unix to make OSX operating system. |
| GUI | Linux is command based but some distros provide GUI based Linux. Gnome and KDE are mostly used GUI. | Initially it was command based OS, but later Common Desktop Environment was created. Most Unix distributions use Gnome. |
| Interface | The default interface is BASH (Bourne Again SHell). But some distros have developed their own interfaces. | It originally used Bourne shell. But is also compatible with other GUIs. |
| File system support | Linux supports more file system than Unix. | It also supports file system but lesser than Linux. |
| Coding | Linux is a Unix clone,behaves like Unix but doesn't contain its code. | Unix contain a completely different coding developed by AT&T Labs. |
| Operating system | Linux is just the kernel. | Unix is a complete package of Operating system. |
| Security | It provides higher security. Linux has about 60-100 viruses listed till date. | Unix is also highly secured. It has about 85-120 viruses listed till date |
| Error detection and solution | As Linux is open-source,whenever a user post any kind of threat, developers from all over the world start working on it. And hence, it provides faster solution. | In Unix, users have to wait for some time for the problem to be resolved. |

**LINUX DISTRIBUTIONS**

**What are Linux Distributions?**

As we all know that Linux is an Open-Source OS used by programmers, organizations, profit and non-profit Organizations around the world in Order to create the custom Operating system to suit their Individual requirement.

The Versions types or kinds of Linux is called as Distributions or Distros.

**How Many Distributions are there?**

There is 100+ Distributions available these days.

Many of them are built for a specific purpose like to run a web server for a network switches like Router or Modems

One of the most popular smart phone-based Linux Distributions is ANDROID.

**Some Examples of Linux Distributions**

**-ARCH LINUX -** Popular Among developers à it is an independent system àIt is designed for a person who goes with Do-It-Yourself approach.

>Developers love it.

>DIY Approach

**-CENTOS -**  Most used for enterprises and web servers à It is a free enterprise class OS and Based on REDHAT ENTERPRISE DISTRO.

**-GENTO -** Source Based Distributions, which means you need to configure the code on your system before you install it à It is not for beginners but for experience users.

**-LINUX Mint -** It is the most popular Desktop Distributions à Launched in 2006 à it is the fourth most used operating system in the computing world.

**-UBUNTU -** Third most popular Desktop Operating System After Microsoft Windows and Apple MAC OS à It is based on Debian Linux Distributions and it is known for its Desktop Environment.

Each Linux Distributions is built for a specific purpose to the demands of its target users.

We can choose any Distributions but as a beginner we’ll start with ubuntu because it is very easy to use and to understand.

**BENEFITS OF LINUX**

* Famous among programmers as well as regular computer usrs around the world.
* Free operating system - no need of spending lot of money like windows.
* Being open source , anyone with programming knowledge can modify it.
* The Linux Operating system now offer millions of Programming tools and applications to choose from , most of them free!
* Once you have installed Linux in your system you no longer , need an antivirus. The main reason is that Linux Operating system is virus free , is due to fact that .exe files do not work on Linux Operating System . All of major viruses either worms, trojans horses etc all of them are found in .exe format only. Thus, the viruses work easily on Windows Platform But they fail in Linux Environment.

**Boot order:**

-> BIOS (Basic input and output system)

-> MBR

-> GRUB (select kernel)

-> Systemd (mother of all process)

->Runlevel 0 - Poweroff/on

-> 1 – rescue

-> 2 - Non-Gui

-> 3 – CLI

-> 4 – unused

-> 5 - Graphical Interface(X windows System)

**Manual partition:**

UEFI System:(Your partition must be GPT)

----------------

EFI – 500MB

swap – 4098MB (2\*RAM)

/ – (15GB)

/home - Remaining space

Legacy BIOS:( Your partition must be MBR)

----------------

/boot – 500MB

swap – 4098MB (2\*RAM)

/ – (10GB)

/home - Remaining space

**LINUX BOOTING SEQUENCE**

**1.BIOS**

>BIOS stands for Basic Input/Output System

>Performs some system integrity checks

> Searches, loads, and executes the boot loader program.

> It looks for boot loader in floppy, cd-rom, or hard drive

> Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.

>So, in simple terms BIOS loads and executes the MBR boot loader.

**2. MBR**

> MBR stands for Master Boot Record.

> It is located in the 1st sector of the bootable disk.

> MBR is less than 512 bytes in size.

>This has three components

1) primary boot loader info in 1st 446 bytes

2) partition table info in next 64 bytes

3) mbr validation check in last 2 bytes.

>It contains information about GRUB (or LILO in old systems).

>So, in simple terms MBR loads and executes the GRUB boot loader.

**3. GRUB**

> GRUB stands for Grand Unified Bootloader.

>If you have multiple kernel images installed on your system, you can choose which one to be executed.

>GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.

>GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).

> GRUB only loads and executes images from Kernel and Initrd.

**4. Kernel**

>Mounts the root file system as specified in the “root=” in grub.conf

>Kernel executes the /sbin/init program

>initrd stands for Initial RAM Disk.

>initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted

**5. Init [RUNLEVEL]**

> Looks at the /etc/inittab file to decide the Linux run level.

Following are the available run levels

Ø 0 – halt

Ø 1 – Single user mode

Ø 2 – Multiuser, without NFS

Ø 3 – Full multiuser mode

Ø 4 – unused

Ø 5 – X11

Ø 6 – reboot

>Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.

>Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level

> If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.

>Typically you would set the default run level to either 3 or 5.

**6. Runlevel programs**

>When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail …. OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.

>Depending on your default init level setting, the system will execute the programs from one of the following directories.

**Install Linux:**

-> load Your pendrive or Virtual Machine

-> choose your language

-> choose keyboard layout

-> choose Ubuntu

-> Setup network (Donot configure DHCP)

-> select use entire disk

-> Select a driver

-> confirm partition

-> Setup user profile with password

Boot order:

-> BIOS (Basic input and output system)

-> MBR

-> GRUB (select kernel)

-> Systemd (mother of all process)

->Runlevel 0 - Poweroff/on

-> 1 - rescue

-> 2 - Non-Gui

-> 3 - CLI

-> 4 - unused

-> 5 - Graphical Interface(X windows System)



**FILE STRUCTURES IN LINUX**

**/- root**

**-/bin** – executables

-**/sbin** – executables

**-/etc** – it is very important because it has all the configuration files.

MITM and DOS/DDOS attack we do many configurations so that we change in linux configurations these configurations will be in /etc.

**-/var** – it is also very important

It contains logs , databases ,websites files etc.

**-/temp** – when we reboot the data will get loss but the same file in the var folder when we reboot it will be permanent.

**-/usr** – Details of usr

**-/boot** – contains all boot folders

**-/lib** – Library folders (it contains all the scripts) – nmap scripts

**USERS IN LINUX**

>RUGULAR USER

>ROOT USER

>SERVICE USER

**>REGULAR USER**

-Regular user is created by us while we install ubuntu operating system.

-Your files and folders are saved in /home/<username> which is your home directory.

**>ROOT USER**

-Root user is also created at the time of instructions.

-This user can access restricted files , install softwares and has administrative privileges.

-It is also called as Super user

-Whenever you want to perform any administrative task on linux , you need to login as a root user.

**>SERVICE USER**

-Linux is widely used as a SERVER OPERATING SYSTEM

-Services like Apache , SQUID have their own individual service accounts.

-Having services accounts security of your computer.

**FILE NAME CONVENTION**

In Windows we cannot have with same name of files in the same folder, the second file if it is created with the same same name you will get an error.

While in Linux you can have two files with the same name in the same directory with different cases.

FILE file is not possible in windows

FILE file is possible in Linux

In Linux the directories are created under the home directory.

Now consider a regular user as sowmya123 you can store all the personel files and directories in the directory /home/sowmya123 but you cannot save files outside this user directory and does not have access to the directories of the other user as well.

This concept is similar to the user folder found in windows.

**TERMINAL VS GUI**

Normally in computer we do many task like

BROWSING

CREATING

MOVING AND

DELETING FILES

**HOW TO MANAGE FILES?**

1) CLI – COMMAND LINE INTERFACE – TERMINAL

2) GUI – GRAPHICAL USE INTERFACE – FILE MANAGER

-Commands are flexible and offer more options which is not available in the GUI modes.

-Perform more kwith just one command.

Example : creating an account will undergo there to four steps in GUI , but the same can be done in the Terminal command.

**root@root:~$ sudo usermod -a -G home SOWMYA**

-If you want to move a file from one location to another (multiple files) it will take hours to complete. Whereas , using Regular expression in CLI you can do within second.

**LAUNCHING THE COMMAND LINE INTERFACE IN UBUNTU**

Go to Dash and type terminal

Press CTRL+ALT+T

You will find

**Root@rootvirtualBox:~$**

**Root – Name of the user**

**rootVirtualBox – Computer Name or Host Name**

**: - separator**

**~ - tells that the user works in hoem directory**

**$ - Represents regular user**

**1)PRESENT WORKING DIRECTORY**

To find in which direcoty you are working in , we use a command called **pwd .**

**Pwd** stands for **PRINT WORKING DIRECTORY**

**root@rootVirtualBox:-$ pwd**

**/home/root**

Which tells you are in home directory

**2)CHANGING DIRECTORY**

To move/change from one directory to another directory we go for **cd** command.

**root@rootVirtualBox :~$ cd /tmp**

**root@rootVirtualBox :/temp$ cd /bin**

**root@rootVirtualBox :/bin$ cd /tmp**

**root@rootVirtualBox :/temp$**

**3)NAVIGATING TO HOME AND ROOT DIRECTORY**

cd ~ - Home directory

cd / -Root Directory

**root@rootVirtualBox :/temp$ cd ~**

**root@rootVirtualBox :~$ -------------------- > Home Directory**

**root@rootVirtualBox :~$ cd /**

**root@rootVirtualBox : /$ --------------------- > Root Directory**

**4)NAVIGATING MULTIPLE DIRECTORIES**

By specifying its complete path

If you want to move dev/bus , we need not break this operation into two different commands like

Cd /dev

Cd /bus

**root@rootVirtualBox :~$ cd /dev/bus**

**root@rootVirtualBox : /dev/bus$**

**Moving up one directory level**

We use a command – cd ..

**root@rootVirtualBox :~$ cd /dev/bus**

**root@rootVirtualBox : /dev/bus$ cd ..**

**root@rootVirtualBox : /dev$ cd ..**

**root@rootVirtualBox :~$**

**PATH AND ITS TYPES**

A path in computing is the address of a file or folder

Eg: In windows : c:\Documents\user\Downloads -------> in windows

In Linux : /home/user/Downloads

**TYPES OF PATH**

> Absolute Path

> Relative Path

**1)Absolute Path**

If we want to move to picture directory to search of image we should give the absolute path to reach the directory.

**root@rootVirtualBox :~$ cd /home/sowmya/Pictures**

**root@rootVirtualBox :~ /Pictures$**

This is the Absolute Path

**2)Relative Path**

Comes when you want to Browse an other Sub Direcotry

Save us from the effort to type the complete paths all the time

Suppose if you are in the Home directory and you want to navigate the pictures directory , for this no need to type absolute path

**root@rootVirtualBox :~$ pwd**

**/home/root**

**root@rootVirtualBox :~$ cd Pictures**

The Path to reach an already open directory.

**IMPORTANT LINUX COMMANDS**

**->LISTING FILES**

To check the list of files in Linux System we use a command called **ls.**

Displays files in the current directory.

**root@rootVirtualBox :~$ ls**

**Documents Downloads Music Pictures Public**

**#NOTE#**

Directories are denoted in Blue color

Files are denoted in white color

Config files are denoted in red color

The ls command will show all the files in the current directory , if suppose you want to check the presence of mp3 file inside music folder under sub directory English and the subdirectory HARDROCK Just give a command ls -R

We use ls -R will show all the files and folders not only in the directory but also the subdirectories.

**root@rootVirtualBox :~$ ls -R**

**./Desktop :**

**./Documents :**

**./Downloads :**

**./Music:**

English:

./Music/English:

Hardrock Pop

./Music/English/Hardrock:

To get a detailed information related to the files and directories use , ls -al or ls -l

**root@rootVirtualBox :~$ ls -al**

**drwxrwxr-x 2 root root 4096 may 6 19:35 batch67**

**drwxrwxr-x - File types and permissions**

**2 - Memory Block occupied by the directory**

**Root -owner of the file**

**Root -Owner of the file or group**

**4096 - File Size in bytes**

**May 6 19:35 - Date and Time**

**Batch67 - Directory or file**

**-VIEWING HIDDEN FILES**

Hidden files start from ‘.’ Period Symbol

To view the Hidden Files use ls -a

**root@rootVirtualBox :~$ ls -a**

**-CREATING OR VIEWING FILES:**

We use CAT command to view a file

We also use it to

>Display

>Copy

>Combine

>Create

**TO CREATE A FILE**

**root@rootVirtualBox :~$cat >file1**

**this is the test file**

Once done giving the content to the file just enter D to return to the command prompt.

This is hown we create a file using CAT Command

**TO VIEW THE SAME FILE**

**root@rootVirtualBox :~$cat file1**

**this is the test file**

**Then create another file**

**root@rootVirtualBox :~$cat >file2**

**this Is the second test file**

**COMBINE TWO FILES**

**root@rootVirtualBox :~$cat file1 file2 >newfile**

Then the files are concatenated

You will not get any conformation message because the bash is silent

It will only shows the message if something goes wrong and error has occurred.

**root@rootVirtualBox :~$cat newfile**

Only the test files can be combined and displayed using cat command.

**DELETING FILES:**

To delete the files that we do not need is done by **rm** Command.

**Syntax: rm <filename>**

**root@rootVirtualBox :~$ rm file1**

The **file1**  will be deleted.

**MOVING FILES**

To move the file to a new location we use **mv** command.

**Syntax: mv <filename><new filelocation>**

**root@rootVirtualBox :~$ mv test /home/root/Pictures**

The move command need the administrative privilege (super user) because currently we are in the standard user and we do not have a right permission for the files / Directories.

To overcome the error just use sudo before mv.

**THE SUDO COMMAND**

The sudo command allows regular users to run programs with security previledges of super user or root user.

**RENAMING FILES:**

To rename a file use the same **mv** command.

**Syntax: mv <filename><new filename>**

**root@rootVirtualBox :~$ ls**

**Documents Downloads file1 file2 test Pictures**

**root@rootVirtualBox :~$mv test TEST**

**root@rootVirtualBox :~$ ls**

**Documents Downloads file1 file2 TEST Pictures**

**MANIPULATING DIRECTORIES**

**>Creating Directories**

To create sub directory in your present working Directory (/home directory)

**Syntax:mkdir <directory name>**

**root@rootVirtualBox :~$ mkdir songs**

**>Creating directory at a new location**

To create a directory at a new location otherthan home directory you need to use **mkdir** command.

**Syntax:mkdir </path/new\_directory\_path>**

**root@rootVirtualBox :~$ mkdir /temp/Music**

**>Creating multiple directories**

To create a multiple directory run this command

**Syntax:mkdir <dir1> <dir2> <dir3>**

**root@rootVirtualBox :~$ mkdir direc direc1 direc2**

**root@rootVirtualBox :~$ ls**

**direc direc1 direc2**

**>Removing Directories**

In order to remove the directory use the command **rmdir.**

**Syntax:rmdir <directory name>**

**root@rootVirtualBox :~$ rmdir direc1**

Delete all the subdirectories and files inside the direc1 and then delete this directory.

>**Renaming the Directory**

To rename the directory use the same **mv** command.

**Syntax: mv<directoryname><new directoryname>**

**root@rootVirtualBox :~$ mv direc2 direc1**

**THE MAN COMMAND**

>Man stands for manual

>The reference book for linux

>It is similar to **HELP**

**>**If you do not understand any command go with manual command and the command name.

**root@rootVirtualBox :~$ man ls**

**Will display all the details of ls command.**

**HISTORY COMMAND**

This history commands shows commands used in the past.

This can help you to refer the old commands and reuse them.

**root@rootVirtualBox :~$ history**

**cd**

**cd file1**

**ls**

**man ls**

**CLEAR COMMAND**

The clear command is used to clean the terminal.

**root@rootVirtualBox :~$ clear**

To copy anything from editor and paste it in the terminal **ctrl+c**

To paste it in Terminal **ctrl+shift+v**

**Pipe, Grep and Sort Command**

>What is a Pipe in Linux?

>'pg' and 'more' commands

>The 'grep' command

>The 'sort' command

>What is a Filter?

**>What is a Pipe in Linux?**

The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next.

In short, the output of each process directly as input to the next one like a pipeline.

The symbol '|' denotes a pipe.

Pipes help you mash-up two or more commands at the same time and run them consecutively.

When you use 'cat' command to view a file which spans multiple pages, the prompt quickly jumps to the last page of the file, and you do not see the content in the middle.

To avoid this, you can pipe the output of the 'cat' command to 'less' which will show you only one scroll length of content at a time.

cat filename | less

An illustration would make it clear.

'pg' and 'more' commands

Instead of 'less', you can also use.

cat Filename | pg

or

cat Filename | more

And, you can view the file in digestible bits and scroll down by simply hitting the enter key.

The 'grep' command

Suppose you want to search a particular information the postal code from a text file.

You may manually skim the content yourself to trace the information. A better option is to use the grep command. It will scan the document for the desired information and present the result in a format you want.

**Syntax:**

**grep search\_string**

Here, grep command has searched the file 'sample', for the string 'Apple' and 'Eat'.

Following options can be used with this command.

Option Function

-v Shows all the lines that do not match the searched string

-c Displays only the count of matching lines

-n Shows the matching line and its number

-i Match both (upper and lower) case

-l Shows just the name of the file with the string

Let us try the first option '-i' on the same file use above -

Using the 'i' option grep has filtered the string 'a' (case-insensitive) from the all the lines.

The 'sort' command

This command helps in sorting out the contents of a file alphabetically.

**The syntax for this command is:**

**Sort Filename**

Consider the contents of a file.

Using the sort command

There are extensions to this command as well, and they are listed below.

Option

Function

-r Reverses sorting

-n Sorts numerically

-f Case insensitive sorting

The example below shows reverse sorting of the contents in file 'abc'.

What is a Filter?

Linux has a lot of filter commands like awk, grep, sed, spell, and wc.

A filter takes input from one command, does some processing, and gives output.

When you pipe two commands, the "filtered " output of the first command is given to the next.

We have the following file 'sample'

We want to highlight only the lines that do not contain the character 'a', but the result should be in reverse order.

For this, the following syntax can be used.

cat sample | grep -v a | sort - r

**XAMPP:**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages

**-> X - cross platform**

**-> A - apache**

**-> M - mysql**

**-> PP - php, perl**

**WordPress**

WordPress is a free and open-source content management system written in PHP and paired with a MySQL or MariaDB database. Features include a plugin architecture and a template system, referred to within WordPress as Themes.

**TO CONFIG WORDPRESS WITH XAMPP:**

-> Download and install XAMPP

-> Download Wordpress from wordpress.org

-> extract wordpress

->Create a Wordpress folder in C:/XAMPP/htdocs

-> copy all the files in Wordpress downloaded to C:/XAMPP/htdocs/wordpress

->run xampp and search localhost/wordpress in browser

**NGINX SERVER**

**what is a web server?**

>Technically, a web server is any computer that has a running daemon (service) accepting HTTP requests on some port (by default 80), and replying with HTTP responses.

>Serving just HTML (along with CSS, JS, Images) is serving static content. That is content that does not change form one user/condition to another.

>In an internet dominated by dynamic web applications like social networking services, online shopping, banking portals and others, non-static (dynamic) content is generated on the fly and served.

>The web server does this by handling the requests to an application handler (like PHP, Ruby, ASP.net) that will do the necessary processing and handles the web server the final content that will be served to the client.

>A lot of web servers are available in the market today. Microsoft Internet Information Services (IIS), Apache, and Nginx.

**What is Nginx?**

>Nginx is one of the most popular web servers in the world and is responsible for hosting some of the largest and highest-traffic sites on the internet.

>It is more resource-friendly than Apache in most cases and can be used as a web server or reverse proxy.

>Nginx, pronounced eNgineEx, a web server developed by Igor Sysoev in 2002. It was officially released in 2004.

>According to he official website www.nginx.org, it is defined as an HTTP and reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server. As per Netcraft, Nginx has served or proxied more than 28% of the busiest websites in February 2017.

>It is a powerful webserver that is capable of doing lots of complex stuff.

**Why to use Nginx?**

>Speed - Faster page loading

>Acceleration - Performancce acceleration by routing traffic

>Load balancing -

>Scalable concurrent connections handling - Nginx can easily handle increasing number of concurrent requests.

>The ability to operate on relatively cheap hardware

>On-the-fly upgrades

>Ease of installation and maintenance

As a prerequisite for nginx server you must have linux server, non-root user with sudo privileges.

**Step 1 – Installing Nginx**

Update your local package index and install:

**$ sudo apt update**

**$ sudo apt install nginx**

**Step 2 – Adjusting the Firewall**

**$ sudo ufw app list**

**Output:**

**Available applications:**

**Nginx Full**

**Nginx HTTP**

**Nginx HTTPS**

**OpenSSH**

We have to enable the most restrictive profile that will still allow the traffic we have configured, permitting traffic on port 80:

**$ sudo ufw allow 'Nginx HTTP'**

to verify,

**$ sudo ufw status**

**Output:**

**Status: active**

**To Action From**

**-- ------ ----**

**OpenSSH ALLOW Anywhere**

**Nginx HTTP ALLOW Anywhere**

**OpenSSH (v6) ALLOW Anywhere (v6)**

**Nginx HTTP (v6) ALLOW Anywhere (v6)**

**Step 3 – Checking your Web Server**

Check with the systemd init system to make sure the service is running by typing:

**$ systemctl status nginx**

**Output:**

● nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2018-04-20 16:08:19 UTC; 3 days ago

Docs: man:nginx(8)

Main PID: 2369 (nginx)

Tasks: 2 (limit: 1153)

CGroup: /system.slice/nginx.service

├─2369 nginx: master process /usr/sbin/nginx -g daemon on; master\_process on;

└─2380 nginx: worker process

Access the default Nginx landing page to confirm that the software is running properly through your IP address:

In your browser tab: https://your\_server\_ip

**Welcome to nginx!!!**

**Step 4 – Setting Up Server Blocks**

>When using the Nginx web server,

-you can use server blocks (similar to virtual hosts in Apache) to encapsulate configuration details and host more than one domain from a single server.

>We will set up a domain called ABC.com.

**Create a folder, change owner ship, and assing permissions**

**$ sudo mkdir -p /var/www/ABC.com/html**

**$ sudo chown -R $USER:$USER /var/www/ABC.com/html**

**$ sudo chmod -R 755 /var/www/ABC.com**

**Create a sample index.html page using nano or your editor:**

**$ gedit /var/www/ABC.com/html/index.html**

**Sample code:**

<html>

<head>

<title>Welcome to ABC.com!</title>

</head>

<body>

<h1>Success! The ABC.com server block is working!</h1>

</body>

</html>

**Make a new server block:**

**$ sudo gedit /etc/nginx/sites-available/ABC.com**

EDIT the configuration block as below, update our new directory and domain name:

**server {**

**listen 80;**

**listen [::]:80;**

**root /var/www/ABC.com/html;**

**index index.html index.htm index.nginx-debian.html;**

**server\_name ABC.com www.ABC.com;**

**location / {**

**try\_files $uri $uri/ =404;**

**}**

**}**

**Enable the file by creating a link from it to the sites-enabled directory:**

**$ sudo ln -s /etc/nginx/sites-available/ABC.com /etc/nginx/sites-enabled/**

Two server blocks are now enabled and configured to respond to requests based on their listen and server\_name directives:

>ABC.com: Will respond to requests for ABC.com and www.ABC.com.

>default: Will respond to any requests on port 80 that do not match the other two blocks.

To avoid a possible hash bucket memory problem that can arise from adding additional server names, it is necessary to adjust a single value

**$ sudo nano /etc/nginx/nginx.conf**

**uncomment the line:**

**server\_names\_hash\_bucket\_size**

**Test for syntax errors:**

**$ sudo nginx -t**

**Restart Nginx to enable your changes:**

**$ sudo systemctl restart nginx**

Nginx should now be serving your domain name. You can test this by navigating to http://ABC.com

**NIGINX SERVER is installed and configured! ! !**

**After finishing the process we try to load the website but it shows error "Hmm. We’re having trouble finding that site."**

**To rectify the error we need to add the hosts ip address and website name in the hosts text file in /etc/hosts**

**EX: 192.168.1.103 ww.abc.com abc.com and save it...but it do not save because of permission denial**

**it will be saved if u change file permissions for that file**

**$ sudo chmod -R 777 /etc/hosts (changing file permission for content adding)**

**$ change the content of the file as mentioned above**

**$ sudo chmod -R 744 /etc/hosts (file permission as before)**

**Now go to browser and reload your page then it'll shows your page...if not showing restart your ubuntu machine**

**NGINX:**

Nginx, stylized as NGINX or nginx or NginX, is a web server that can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache.

**Serving Files:**

Serving static and websites in nginx

The [root](https://nginx.org/en/docs/http/ngx_http_core_module.html#root) directive specifies the root directory that will be used to search for a file. To obtain the path of a requested file, NGINX appends the request URI to the path specified by the root directive. The directive can be placed on any level within the http {}, server {}, or location {} contexts. In the example below, the root directive is defined for a virtual server. It applies to all location {} blocks where the root directive is not included to explicitly redefine the root:

**EXAMPLE:**

**server {**

**root /www/data;**

**location / {**

**}**

**location /images/ {**

**}**

**location ~ \.(mp3|mp4) {**

**root /www/media;**

**}**

**}**

**Auto Indexing:**

Helps to find index page automatically in a directory

**EXAMPLE:**

**location / {**

**autoindex on;**

**}**

**Creating Directory Listing:**

-> listing files available in servers current directory

To do directory listing configure auto index on and delete the index file in current directory

**Reverse proxy:**

In computer networks, a reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client, appearing as if they originated from the reverse proxy server itself.

**EXAMPLE:**

**server {**

**listen 80;**

**location / {**

**proxy\_pass http://my\_server;**

**}**

**}**

**Custom Config file:**

The default config file for nginx is “nginx.conf” to change the default config go through the below.

● sudo unlink /etc/nginx/sites-enabled/default

● cd /etc/nginx/sites-available/

● sudo vi custom\_server.conf

Replace custom\_server with a name that’s meaningful to you. In the new file, enter:

**server {**

**listen 80;**

**location / {**

**proxy\_pass http://my\_server;**

**}**

**}**

To activate the new Nginx file, enter:

**ln -s /etc/nginx/sites-available/custom\_server.conf**

**Load Balancing:**

In computing, load balancing refers to the process of distributing a set of tasks over a set of resources, with the aim of making their overall processing more efficient.

**http {**

**upstream backend {**

**server backend1.example.com weight=5;**

**server backend2.example.com;**

**server 192.0.0.1 backup;**

**}**

**server {**

**location / {**

**proxy\_pass http://backend;**

**}**

**}**

**}**

**Logging:**

In nginx we have error log and access log where , error log is used to log run time errors and access log is to log who accessed the file

ERROR LOG EXAMPLE:

**error\_log logs/error.log warn;**

ACCESS LOG EXAMPLE:

**server {**

**access\_log /spool/logs/nginx-access.log;**

**}**

**Blacklist:**

In nginx we can deny particular ip or a network from requesting the nginx server

**EXAMPLE:**

**location /subdirectory/ {**

**deny 1.2.3.4;**

**}**

**Samba Server**

**Step 1: Installing Samba:**

**----------------------------**

To install Samba, we run:

$ sudo apt update

$ sudo apt install samba

We can check if the installation was successful by running:

$ whereis samba

**Step 2 : Setting up Samba**

**-----------------------------**

Now that Samba is installed, we need to create a directory for it to share:

mkdir /home/<username>/sambashare/

The configuration file for Samba is located at /etc/samba/smb.conf. To add the new directory as a share, we edit the file by running:

$ sudo gedit /etc/samba/smb.conf

The file \*smb.conf\* is divided in several sections:

\*Global Settings

\*Debugging/Accounting

\*Authentication

\*Printing

\*File sharing

\*Misc

\*Share Definitions

**File Sharing (Basics)**

**----------------------**

The important part for us is File sharing. Samba shares are named in brackets, [ ], and configured by adding options in the lines that follow. Most options are boolean (yes / no).

We need to change:

[homes]

comment = Home Directories

browseable = no

**File Sharing (Advanced)**

**---------------------------**

If you have more than one network card (or interface) then you have to define where you want Samba to run.

In smb.conf under the [global] section, add:

interfaces = 127.0.0.1, 192.168.0.31/24

bind interfaces only = yes

You can limit which IP address can connect to your Samba server adding these lines:

hosts allow = 127.0.0.1, 192.168.0.31, 192.168.0.32

hosts deny = 0.0.0.0/0

The loopback address must be present in the first line. The second line deny access from all IP address not in the first line.

**Private and public shares in same config**

**---------------------------------------------**

First you'll want to set this up in the [global] section of your smb.conf

[global]

security = user

encrypt passwords = true

map to guest = bad user

guest account = nobody

Next the private share

[private]

comment = Private Share

path = /path/to/share/point

browseable = no

read only = no

Finally, the public share

[public]

comment = Public Share

path = /path/to/share/point

read only = no

guest only = yes

guest ok = yes

**At the bottom of the file, add the following lines:**

**------------------------------------------------------**

[sambashare]

comment = Samba on Ubuntu

path = /home/username/sambashare

read only = no

browsable = yes

guest ok = yes

create mask = 0644

directory mask = 0755

What we’ve just added

\*comment: A brief description of the share.

\*path: The directory of our share.

\*read only: Permission to modify the contents of the share folder is only granted when the value of this directive is no.

\*browsable: When set to yes, file managers such as Ubuntu’s default file manager will list this share under “Network” (it could also appear as browseable).

---------------------------------------

Add users who can access your shares with the 'smbpasswd' command.

$ sudo smbpasswd -a username

New SMB password:

Retype new SMB password:

Added user username.

$ sudo smbpasswd -e username

Enabled user username.

NOTE:

The username used here should be a real user setup on your PC/Server. Reload Samba for every change to users/passwords or 'smb.conf'

$ sudo /etc/init.d/samba reload

NOTE:

If the above command doesn't work for you, try:

$ sudo smbd reload

Now that we have our new share configured, save it and restart Samba for it to take effect:

$ sudo service smbd restart

Update the firewall rules to allow Samba traffic:

$ sudo ufw allow samba

Now from client machine from run command search for the server\_ip to access the shared folders:

\\ip-address\sambashare

Samba server is configured! ! !

FTP Server

>File Transfer Protocol (FTP) is a TCP protocol for downloading files between computers.

>In the past, it has also been used for uploading but, as that method does not use encryption, user credentials as well as data transferred got easily intercepted.

>FTP works on a client/server model.

>The server component is called an FTP daemon.

>It continuously listens for FTP requests from remote clients.

>When a request is received, it manages the login and sets up the connection.

>For the duration of the session it executes any of commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

\*Anonymous

\*Authenticated

Step 1: vsftpd - FTP Server Installation

-------------------------------------------

$ sudo apt install vsftpd

$ sudo systemctl start vsftpd

$ sudo systemctl enable vsftpd

Step 2: Configure Firewall

----------------------------------

$ sudo ufw allow 20/tcp

$ sudo ufw allow 21/tcp

Step 2: Anonymous FTP Configuration

-------------------------------------------

By default vsftpd is not configured to allow anonymous download. If you wish to enable anonymous download edit /etc/vsftpd.conf by changing,

anonymous\_enable=Yes

During installation a ftp user is created with a home directory of /srv/ftp. This is the default FTP directory.

To change this directory create a new directory,

$ sudo mkdir /srv/files/ftp

$ sudo usermod -d /srv/files/ftp ftp

After making the change restart vsftpd:

$ sudo systemctl restart vsftpd.service

Finally, copy any files and directories you would like to make available through anonymous FTP to /srv/files/ftp, or /srv/ftp, if you wish to use the default.

Step 3: User Authenticated FTP Configuration

----------------------------------------------------

>By default vsftpd is configured to authenticate system users and allow them to download files.

>If you want users to be able to upload files, edit /etc/vsftpd.conf:

write\_enable=YES

Now restart vsftpd:

$ sudo systemctl restart vsftpd.service

Similarly, by default, anonymous users are not allowed to upload files to FTP server.

To change this setting, you should uncomment the following line, and restart vsftpd:

anon\_upload\_enable=YES

Enabling anonymous FTP upload can be an extreme security risk.

It is best to not enable anonymous upload on servers accessed directly from the Internet.

FTP Server is installed and configured! ! !

OUTPUT VERIFICATION:

Open command prompt from windows

Just type ---> ftp <ip of ftp server>

next give the username of ftp server

and then give the password of the server

ftp> ls

list out all the directories

DOWNLOAD FILE FROM FTP SERVER

CRATE A FTP FOLDER AND SAVE SOME FILE EXAMBLE 1.TXT 2.TXT 3.TXT

SAVE.

OPEN COMMENT PROMT CONNECT FTP 192.168.43.172

MGET-DOWNLOAD MULTIPLE FILE FROM SERVER

GET-DOWNLOAD SINGLE FILE

CHOSSING DIR FROM DOWNLOAD FILE USING COMMENT

FTP>lcd c:\shareftp

THAN USING

ftp>cd ftp

ftp>mget \*

PUT COMMENT USED TO UPLOAD A FILES

ftp> cd ftp

ftp> pwd

257 "/home/sathish/ftp" is the current directory

ftp> put c:\shareftp\sathish

200 PORT command successful. Consider using PASV.

150 Ok to send data.

226 Transfer complete.

ftp: 199 bytes sent in 0.00Seconds 199.00Kbytes/sec.

VERYFING SERVER IN FTP FOLDER