**LINUX**

Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged as a Linux distribution, which includes the kernel and supporting system software and libraries, many of which are provided by the GNU Project.

**Open-Source:**

Open-source software is software with source code that anyone can inspect, modify, and enhance. By design, open-source software licenses promote collaboration and sharing because they permit other people to make modifications to source code and incorporate those changes into their own projects. They encourage computer programmers to access, view, and modify open-source software whenever they like, as long as they let others do the same when they share their work.

**History of Linux:**

* Linus Torvalds identified few issues in Unix.
* **Linus Torvalds** started developing his own Operating System by using 'Minux' OS

(Li)nus Toravalds + Mi(nux) ==> Linux OS

* Linus Torvalds released Linux OS for free of cost.
* Linux Torvalds provided Linux OS source code also in the internet
* People and companies downloaded source code of Linux OS and they modified according to their requirement and released into market with their brand names
* As companies released their own Linux OS we can see several distributions for Linux OS

Ex:

* + - 1. Ubuntu
      2. CentOS
      3. RedHat
      4. Fedora
      5. SUSE
      6. KALI
      7. Debian etc....

**Note: There are 200+ Linux Distributions available in the market**

**Architecture:**

The architecture of a Linux System consists of the following layers –

* **Hardware layer** − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
* **Kernel** − It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
* **Shell** − an interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
* **Applications/ Utilities** – Applications/Utility programs that provide the user most of the functionalities of an operating systems.



**Features of Linux:**

Following are some of the important features of Linux Operating System.

* **Portable** − Portability means software can works on different types of hardware in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
* **Open Source** − Linux source code is freely available and it is community-based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
* **Multi-User** − Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
* **Multiprogramming** − Linux is a multiprogramming system means multiple applications can run at same time.
* **Hierarchical File System** − Linux provides a standard file structure in which system files/ user files are arranged.
* **Shell** − Linux provides a special interpreter program which can be used to **execute commands** of the operating system. It can be used to do various types of operations, call application programs. etc.
* **Security** − Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

**Environment Setup:**

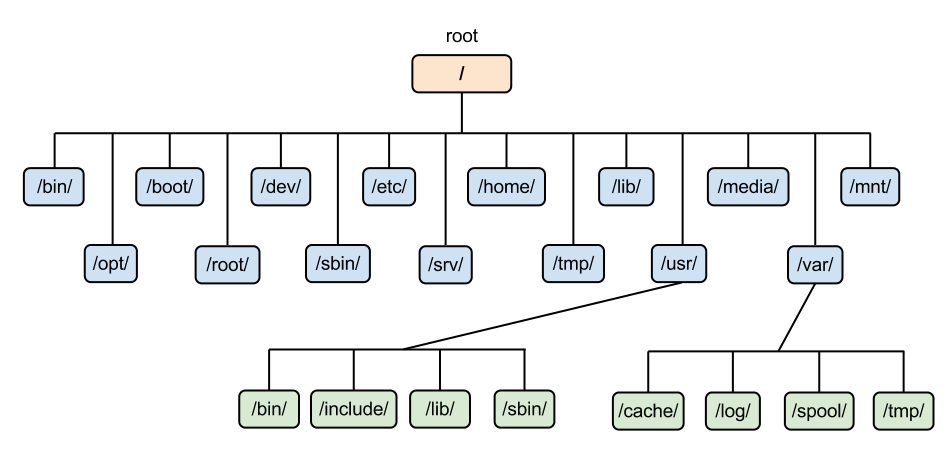
1) Create Account in AWS (Free Tier)

2) Create Linux Virtual Machine in AWS using EC2 service

3) Connect to Linux Virtual Machine using MobaXterm / Putty

**Linux File System:**

Linux file system is the hierarchical directory structure used by Linux-based operating systems to organize and store files and directories. The file system in Linux is represented as a tree structure, with the root directory at the top, and other directories and files branching off from it.



The Linux file system follows the Filesystem Hierarchy Standard (FHS), which specifies the organization of files and directories on a Linux system. The FHS defines several important directories such as:

/ : The root directory and the starting point of the file system.

/bin : Contains essential system binaries such as ls, cat, cp, mv, etc.

/boot : Contains the files required for booting the system, including the Linux kernel, initrd images, and boot loader configuration files.

/dev : Contains device files, which are used to communicate with hardware devices.

/etc : Contains system-wide configuration files such as network settings, user accounts, and system services.

/home : Contains user home directories.

/lib : Contains system libraries required by programs on the system.

/mnt : Mount point for temporary file systems such as external hard drives, USB drives, etc.

/opt : Contains optional software packages that are not part of the core system.

/proc : Contains virtual files that provide information about the system's hardware and running processes.

/root : Home directory for the root user.

/sbin : Contains system binaries used for system administration tasks.

/tmp : Directory for temporary files.

/usr : Contains files related to user applications, including binaries, libraries, documentation, and source code.

/var : Contains variable files such as log files, mail spools, and temporary files generated by system processes.

In addition to these directories, there are other directories and files that are used for specific purposes on a Linux system, and the user can also create their own directories and files to organize their data.

**Types of files:**

In Linux, there are many types of files that serve different purposes. Here are some of the common types of files in Linux:

* **Regular files**: These are the most common type of files in Linux. They can contain text, images, audio, video, or any other type of data.
* **Directories**: Directories are used to organize files into a hierarchical structure. They can contain regular files, other directories, or special files.
* **Symbolic links**: Also known as soft links, symbolic links are files that point to another file or directory. They can be used to create shortcuts or to refer to files that have been moved or renamed.
* **Block devices**: These files represent devices that store data in blocks, such as hard drives or USB drives.
* **Character devices**: These files represent devices that read or write data character by character, such as keyboards or printers.
* **Named pipes**: Also known as FIFOs (First In, First Out), named pipes are used for inter-process communication.
* **Sockets**: Sockets are used for inter-process communication between different computers or different processes on the same computer.
* **Special files**: Special files are used to provide access to system resources, such as /dev/null, which is used to discard data, or /dev/random, which provides random data.

These are just a few of the common types of files in Linux. There are many other types of files used for various purposes, such as configuration files, log files, and more.

**Files creation:**

**touch**: It is used to create empty files

**Syntax:**

touch <filename>

touch f1.txt

touch a1.txt a2.txt a3.txt

**cat** : It is used to create files with content & it is used to print file content

**Syntax:**

cat > filename : Create file with data (ctrl + d to close the file)

cat >> filename : Append data to existing file

cat filename : Print file content

cat filename1 > filename2 : Copy data from one file to another file

cat f1.txt f2.txt > f3.txt : Copy more than one file data into another file

**mv** : It is used to move/rename the files

Syntax:

mv present-name new-name

**rm** : It is used to remove the files

syntax:

rm filename

**Working with Directories:**

**mkdir** : it is used to create/make directory

rmdir : it is used to remove only empty directory

**rm -r <dirname>** : It is used to delete non-empty directories

**cd <dirname>** : Change directory

**cd ..** : Come out from the directory

**ls -l <dirname>** : list the content of given directory

**cat <filename>** : It will display all the data available in the file

**head** : It will display first 10 lines of the file from top (10 is the default count)

head <filename> (it will give first 10 lines of data)

head -n 15 <filename> (it will give first 15 lines of data)

head -n 25 <filename> (it will give first 25 lines of data)

**tail** : It will display last 10 lines of the file from bottom (10 is the default count)

tail <filename> (it will give last 10 lines data)

tail -n 15 <filename> (it will give last 15 lines data)

tail -n 25 <filename> (it will give last 25 lines data)

tail -n 50 <filename> (it will give last 50 lines data)

tail -n +50 <filename> (it will give data from 50th line to till last line)

Note: To get latest data from file we need to use 'tail' command because lastest data will be appended at bottom

**grep** : Global Regular Expression Print

**Grep is used for searching**

grep -i 'Linux' \* (It will search for linux word in all the files)

grep -i 'Linux' <filename> (it will search for linux word in given filename)

grep –n ‘Linux’ <filename> (to show linu number where it is found)

**wc** : (word count) it is used to count no.of lines, words and no.of characters of given file

wc <filename>

**cp** : It is used for copy and paste

cp <oldfilename> <newfilename>

Note: If we want to copy more than one file data then we should go for 'cat' command

cat f1.txt f2.txt > f3.txt

**mv** : it is used for renaming & moving

mv <existing-file-name> <new-file-name>

mv <source> <destination>

**Text Editor**

vi : Visual Editor

**vi <filename>** : It will open the file

1. press 'i' to enter into 'insert' mode
2. in insert mode we can change file data
3. press 'esc' to come out from insert mode
4. press :wq to save and quit the file
5. press :q! to close the file without saving

**SED : Stream Editor**

‘sed’ command in linux stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. By using ‘sed’ you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it. This command supports regular expression which allows it perform complex pattern matching.

Examples

1. **sed 's/text1/text2/' filename** (find ‘text1’ and replace with ‘text2’. First occurrence in each line)

2.  **sed 's/text1/text2/2' filename** (find ‘text1’ and replace with ‘text2’ for second occurrence in every line)

3. **sed 's/text1/text2/g' filename** (find ‘text1’ and replace with ‘text2’ for all occurrences of word ‘text1’)

4. **sed 's/text1/text2/3g' filename** (find ‘text1’ and replace with ‘text2’ from 3rd occurrence in each line)

5. **sed '3 s/text1/text2/' filename** (replace ‘text1’ with ‘text2’ on 3rd line first occurrence)

6. **sed 's/text1/text2/p' filename** (to print the line on which text1 is replaced twice and all other lines only once)

7. **sed -n 's/text1/text2/p' filename** (prints only the lines on which the text is replaced and ignores other lines)

8. **sed '1,3 s/text1/text2/' filename** (replaces the text once between line 1 and line 3 only)

9. **sed '5d' filename** (deletes line 5 in the file without opening it)

10. **sed ‘$d’ filename or sed ‘^d’ filename** (delete last or first line of the file respectively)

11. **sed ’10,$d’ filename** ( delete all lines from 10th line to end of file)

12. **sed '/^$/d' filename** (delete all blank lines from the file)

13. **sed '/pattern/d' filename** (delete the line that matches the pattern)

14. **sed 'G;G' filename** (insert two blank lines after each line in the file, one line is inserted for each ‘G’ in the command)

15. **sed 's/^/     /' filename** (insert 5 space indentation to every line in file)

16. **sed '/./=' filename | sed '/./N; s/\n/ /'** (insert line number for each line only if the line is not blank)

17. **sed -i '/^#/d;/^$/d' filename** (deletes lines that begin with # and also blank lines)

18. **sed -n '2,5p' filename** (view particular lines of a file, line 2-5 in this case)

**The man command in Linux is used to display the manual pages (short for "manuals") for a particular command or program.**

**man ls**

**man cat**

**man tac**

**man : manual pages of command**

**To read details/description about command**

**Steps:**

**1. man ls (it opens ls command manual page)**

**2. Arrow keys to scroll (Up / down)**

**3. come out of manual page 'q'**

**4. To search any word press '/' then word to search**

**ex: /--all**

**5. click 'n' to go to next matching**

**click 'N' (or shift+n) to goto previous matching**

**6. 'g' to come to starting of the page**

**'G' to goto last line of the page**

**Find & Locate Commands**

* find & locate commands are used for file search in linux machine
* locate command will perform search operation in locate database (internal database in linux)
* find command will perform search operation in entire linux file system (It will give accurate result)
* **locate apache** (it will give all the files path which are having apache in the file name)
* **locate -c apache** (it will give file count)
* **find /home -name python.txt** (it will search for the file with name python.txt inside /home directory)
* **find /home -type f -empty** (It will print all empty files available inside /home directory)
* **find /home -type d -empty** (it will print all empty directories)

Examples:

**find <dir> -name filename** (find a specific file)

**find <dir> -iname filename** (ignore case in filename)

**find <dir> -type d -name dirname** (find directories by name)

**find <dir> -type f -name ‘\*.txt’** (find files of specific extension)

**find <dir> -type f -perm 775** (find files with specific permission type)

**find <dir> -type f ! -perm 775** (find files which are not specific permission type)

**find <dir> -type f -perm 777 -exec chmod 644 {} \;** (find files with permission levels and change permission level)

**find <dir> -type f -name ‘\*.txt’ -exec rm -I {} \;** (find and delete files of specific extension)

**find <dir> -type f -empty** (find all empty files in the directory)

**find <dir> -type d -empty** (find empty sub-directories within the directory)

**find <dir> -type f -name ‘.\*’** (finds all hidden files)

**find <dir> -type f -perm u=x** (finds all executable files)

**find <dir> -type f -user username** (finds all files to which username is owner)

**find <dir> -type f -group groupname** (finds all files to which groupname is owner)

**find <dir> -mtime 5** (finds all files that were modified in last 5 days)

**find <dir> -atime** **2** (finds all files that were accessed by a user in last 2 days)

**find / -type f -size +100M -exec rm -f {} \;** (finds and delete all files greater than 100 mb)

**History:**

The history command in Linux is used to display a list of previously executed commands from the user's command-line history. The command history is stored in a file called .bash\_history in the user's home directory.

The general syntax of the history command is:

**history [option(s)]**

By default, the history command will display the last 500 commands executed in the current shell session, along with line numbers. You can use the -n option followed by a number to display a different number of commands, for example:

**history -n 1000**

This will display the last 1000 commands executed in the current shell session.

You can also use the history command in conjunction with other commands. For example, you can use the grep command to search for a specific command in the history:

**history | grep command**

This will display all commands in the history that contain the word "command".

Finally, you can use the ! operator to execute a specific command from the history. For example, to execute the 10th command from the history, you can type:

**!10**

This will execute the 10th command from the history.

**history: It will give commands execution history**

**Note:**

**history**

**history –c : to clear history commands**

**!! to execute last command**

**!<number> to repeat the command from hostory**

**Working with User Accounts in Linux**

1. Linux is a multi-user operating system
2. Within one Linux machine we can create multiple user accounts
3. Multiple users can access one linux vm at a time and they can perform Multi-tasking
4. Note: Every linux machine will have 'root' account (super user --> sudo )
5. When we launch a linux vm using 'Amazon Linux AMI' we will get 'ec2-user' account by default

**whoami:** it will give currently logged in username

**id** <username> : It will give information about user account

**pwd** : It will display present working directory

Note: For every user account, one home directory will be created.

For ec2-user account the home directory is : /home/ec2-user

# Switch to root user

$ sudo su – (or) $sudo su

Note: To perform admin activities we will use 'sudo' permission

**User/group commands:**

# create user

**$ sudo useradd <username>**

# set password for user

**$ sudo passwd <username>**

# verify user account details

**$ id <username>**

# List all users in Linux

**$ cat /etc/passwd**

# switch user account

**$ su <username>**

Note: After swithcing username, change home directory also

# delete user

**$ sudo userdel <username>**

# Display all groups

**$ cat /etc/group**

# Create a group

**$ sudo groupadd <groupname>**

# Add user to a group

**$ sudo usermod -aG <group-name> <user-name>**

# Remove user from the group

**$ sudo gpasswd -d <username> <group-name>**

# Delete group

**$ sudo groupdel <group-name>**

# print users who are belongs to particular group

**$ sudo lid -g <group-name>**

**File Permissions**

File Permissions will decide who can do what on that file. Every File will have 3 types of permissions

1) Read (r)

2) Write (w)

3) Execute (x)

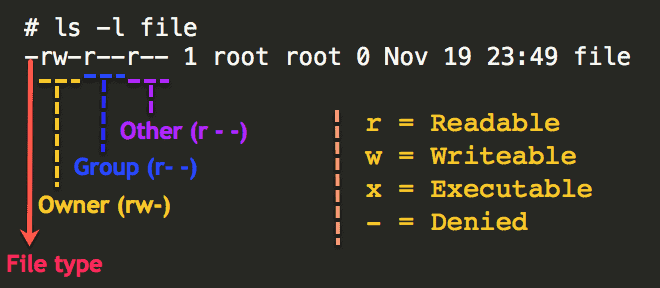
In Linux, File Permissions are divided into 3 sections

1) User Permissions (first 3 characters)

2) Group Permissions (charters 4, 5, 6)

3) Others Permissions (charaters 7,8 & 9)

Ex: rwxrwxrwx



To change file permissions we have 'chmod' command

# add execute permission for user

**$ chmod u+x <filename>**

# add execute permission for group

**$ chmod g+x <filename>**

// remove write permission for group

**$ chmod g-w <filename>**

// add write & execute permissions for others

**$ chmod o+wx <filename>**

// add write permissions for others (2 files at a time)

**$ chmod o+w <file1> <file2>**

**We can represent file permissions with Numeric Numbers also,**

r => Read (4) w => write (2) and x = execute (1)

**$ chmod 777 <filename>**

**$ chmod 765 <filename>**

**chown :** 'chown' command is used to change file ownership

We can see owner of the file using 'ls -l' command

# changing owner of a file

**$ sudo chown <uname> <filename>**

We can change file owner using userID also

**$ sudo chown <UID> <filename>**

Note: We can get UID for username using 'id uname' command

$ id ramesh

# change group of a file

$ sudo chown :groupName <fileName>

**Q) What is the difference between 'chmod' and 'chown' commands?**

chmod ---> it is used for managing file permissions

chown ---> it is used for managing file ownership

**Networking Commands**

**ifconfig** : This command is used to get ip address of our machine

**ping** : It is used to check connectivity

$ ping <ip>

0% packet loss means our ping is succesful (we got response from server)

100% packet loss means our ping is failure (no response from server)

**wget** : It is used to download a resource from internet using URL

$ wget <url>

$ wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.zip

Note: We can extract that zip file using 'unzip' command

$ unzip <zip-file-name>

**curl** : It is used to send http request to a server

$ curl <url>

**Package Management / Software Installation in Linux**

To install softwares in Linux we will use Package Manager.

There are several package managers available for Linux.

1) yum

2) apt

3) deb

4) RHEL

Based on Linux distribution we need to choose package manager

**Amazon Linux / Cent OS / Red Hat -------> yum | Ubuntu / Debian ---> apt**

# update existing packages

**$ sudo yum update -y**

# install git client software in linux

**$ sudo yum install git**

# check git version

**$ git --version**

# install java

**$ sudo yum install java**

# install java 1.8 version

**$ sudo yum install java-1.8.0-openjdk**

# check java version

**$ java -version**

# install maven

**$ sudo yum install maven**

# check maven version

**$ mvn -version**

# Un-install software

**$ sudo yum remove maven**

**Hosting Static website in Webserver**

Website nothing but collection of web pages. Websites are 2 types

1) Static website: static website will give same response for user (fixed content)

2) Dynamic website: dynamic website means content will display based on user action/location/time

To run the website we need to deploy that website inside a server

#install webserver

**$ sudo yum install httpd**

Note: httpd is a web server package and it is used for static websites hosting

# start webserver

**$ sudo service httpd start**

Note: HTTPD webserver runs on HTTP protocol with 80 port number (enable that in security group inbound rules)

# modify webpage content

**$ cd /var/www/html**

**$ sudo vi index.html**

Note: write content in index.html file then save and close

# Access website using EC2 VM public IP in your browser

URL: http://13.233.129.51/ | URL: http://13.233.129.51:80/

Note: To deploy & run dynamic websites we will use below servers

1) Apache Tomcat

2) JBOSS

3) Web logic

4) WebSphere

5) Glassfish etc....

**FAQs:**

Q) What is the absolute and relative path?

A) Absolute path always starts from ‘/’ directory to the destination file/folder. Relative path is the path of the destination relative to the present working directory/current directory.

Q) Write a command to list files opened by User.

A) **lsof -u username.**

Q) Number of files in a given directory including sub-directories.

A) **find . type -f | wc -l**

Q) Number of users created in Linux OS.

A) **cat /etc/passwd | wc -l**

Q) Concat 3 files data into a single file.

A) c**at file1 file2 file3 >> destfile**

Q) Write “Hello World” line 10 times into a file.

A) **for i in {1..10}; do echo “Hello World”; done >> filename**

Q) Print line 20 to line 40 of a file.

A) **sed -n ’20,40’p filename**

Q) Replace “Welcome” to “Bye” in a file at all places.

A) **sed -i -E ‘s/(W|w)elcome/bye/g’**

Q) Find count of all files that start with “abc” which are created by sam user.

A) **find . -name ‘abc\*’ -user sam | wc -l**

Q) Compress and zip a directory that contains files to a tarball (.tar.gz)

Create 5 files into directory details

Create tarball details.tar.gz

Extract from the tarball to destination directory “extractedDir”

A) **tar zcvf details.tar.gz details**

**tar zxvf details.tar.gz -C /extractedDir**

**tar tvf filename(to view contents of tarball)**

Q) Move all files to a directory including hidden files.

Create a file ‘.model’, create file ‘format’, create directory ‘one’

Move all files including hidden files to ‘one’ directory.

A) **shopt -s dotglob**

Q) Count number of sub-directories in a given directory.

A) **find . -type d | wc -l**

Q) How to find .txt files and delete with confirmation?

A) **find . -name ‘\*.txt’ -exec rm -I {} \;**

Q) Search for empty files and directories.

A) **find . -empty**

Q) Count the number of times a word is repeated in a file?

A) grep -io expression filename | wc -l

Q) What is xargs in command piping in Linux?

A) **ls | xargs cat**

Q) Display current Linux machine OS details.

A) **cat /etc/os-release**

Q) Display all running process details.

A) **ps -ef**

Q) Kill a process with name of id.

A) **kill <pid> or pkill <processname>**

Q) Display IP address of the machine.

A) **ifconfig | grep “inet” | grep -v “inet6”**

**ip addr show**

**hostname -i**

Q) Display CPU & Memory information.

A) **lscpu or cat /proc/cpuinfo**

**lsmem or cat /proc/meminfo**

Q) How can we encrypt and decrypt file data in Linux?

A) **gpg -c filename (will perform encryption and create filename.gpg file)**

**gpg -d filename.gpg (will perform decryption and create filename file)**

**gpg filename.gpg (same as above decryption)**

Q) How to find machine running time ?

$ uptime: It is used to find from how many hours our machine is running

Q) How to find free space available in linux machine

$ df -h: It is used to find out how much free space available in our machine

Q) How to change password of user ?

$ sudo passwd username

Q) How to display list of users available in linux?

$ cat /etc/passwd : It will print all users available in linux

Q) How to display all groups available in Linux?

$ cat /etc/group : It will print all groups available in linux

Q) How to assign a user for a group?

$ sudo usermod -aG <groupName> <uname>

Q) What is sudo in linux?

Super user (administrator permissions)