

=====
Where we will use Linux OS in Realtime ?
=====

- 1) Application deployment will happen on linux machines only
- 2) Tools will be installed on Linux machines only

Ex: Docker, K8S, Jenkins, Nexus, SonarQube, ELK etc...

- 3) Application log files will be stored in linux machine only

=====
What is OS ?
=====

=> It is a software which acts as mediator between user and computer

=> Users will communicate with computers using OS

=> Without OS we can't use any computer

=> OS provides platform to run our applications in computer

Ex: notepad, calculator, browser, tomcat, eclipse...

=> We have several operating systems in market

Ex: Windows, Linux, Mac, Android, IOS etc...

=====
Windows OS
=====

=> Developed by Microsoft company (Bill Gates)

=> It is commercial OS (licensed)

=> It is single user based OS

=> Security features are very less in Windows

Note: We need to install anti-virus software to protect files

=> Windows is GUI based OS

=> Windows is recommended for personal use only

Ex: internet browsing, games, watch movies, attend online classes....

=====
Linux OS
=====

=> Linux community based OS

=> Linux is free & open source os

=> Linux is CLI based OS

=> Linux is multi user based OS

=> Security Features are very high in linux

Note: Anti-virus s/w is not required

=> Linux OS developed by "Linus Torvalds"

=> Linux is highly recommended to manage servers

Ex: App Servers, DB servers, Docker, Jenkins, K8S etc..

=====

Linux History

=====

-> Linus Torvalds identified some challenges/issues in Unix OS

-> Linus Torvalds identified one OS which is matching with his ideas

i.e Minix OS

-> Linus Torvalds used Minix OS code and made some changes and released into market as new OS

(Li) nux + Mi (nux) ==> Linux

=====

Linux Distributions

=====

-> Linus Torvalds provided Linux OS code for free of cost

-> So many companies downloaded Linux OS source code and modified according to their requirement and released into market with their brand names those are called as Linux Distributions/ Linux Flavours.

-> We have 200+ Linux Distributions in the market.

Ex: Amazon Linux, Ubuntu, CentOS, RedHat, Debian, SUSE, Kali, Fedora....

=====

How to setup Linux Machine ?

=====

Approach-1) Install Linux OS directly in computer

Approach-2) We can use Virtual Box and install Linux OS as Guest OS in Windows

Approach-3) Setup Linux VM in AWS Cloud (Recommended)

=====

Setup Linux VM in AWS Cloud

=====

Step-1 : Login into AWS cloud account

Step-2 : Create EC2 Instance (Linux VM) (Amazon Linux)

instance type : t2.micro (free tier eligible)

Step-3 : Connect with EC2 instance using SSH Client

(git bash/ mobaxterm / putty)

Step-4 : Practice Linux commands

=====

ðŸ’¥ Today's Assignment

=====

ðŸ’% AWS account setup : <https://www.youtube.com/watch?v=xi-JDeceLeI>

ðŸ’% Linux Machine with Git Bash : <https://www.youtube.com/watch?v=Jm1QaTXvw5o>

ðŸ’% Linux Machine with MobaXterm : <https://youtu.be/uI2iDk8iTps?si=ZuZs0lQTxoRpbRMk>

ðŸ’% Linux Machine with putty : https://youtu.be/GXc_bxmP0AA?si=HgSydrP89mPxv23s

=====

Linux Commands

=====

=> Linux is CLI based OS

=> We will perform operations in linux vm using linux commands

whoami : Currently logged user.
pwd (Present working directory)
date: Shows the current date
cal : Shows the current month
cal 2025: Shows 2025 calender

mkdir : Make directory

```
mkdir ashokit
mkdir java python aws devops
```

rmdir : Remove empty directory

```
rmdir devops
```

ls : list the files of present working directory

```
ls -l : Display in alphabetical order
```

```
ls -lr : Display in reverse of alphabetical order
```

```
ls -lt : Display files based on timestamp (latest on top)
```

```
ls -ltr : Display old files on top & latest at bottom
```

```
ls -la : display hidden files (.a)
```

cd : change directory (navigation)

```
cd <dir-name> : go inside the directory
```

```
cd .. : come out from directory
```

touch : To create empty files

```
touch f1.txt
```

```
touch f2.txt f3.txt f4.txt
```

cat : create file with data + append data to file + print file data

```
cat > f1.txt : Create new file with data
```

cat >> f1.txt : Append data to the exiting file data

cat f1.txt : Print file data from top to bottom

cat -n f1.txt : print file data along with line numbers

rm : Remove files

rm f1.txt : Remove the file

rm -rf devops : remove non-empty directory

cp : copy the data from one file to another file

cp f1.txt f2.txt

Note: To copy the data from multiple files we need to use cat command like below

cat f1.txt f2.txt > f3.txt

mv : rename file/directory + move the file/directory

mv linux.txt linux-os.txt

mv git.txt devops/

=====
tac : Read file data from bottom to top (opposite of cat cmd)

tac f1.txt

wc : word count

wc f1.txt

rev : reverse the file data and print it

rev f1.txt

=====
head : To display file data from top (default 10 lines)

head app.log

head -n 20 app.log

head -n 25 app.log

tail : To display file data from bottom (default 10 lines)

tail app.log

tail -n 20 app.log

tail -n 100 app.log

grep : grep stands for global regular expression print

Note : We will use this grep command for keyword search in file

grep 'exception' app.log (print lines which contains given keyword)

grep -i 'exception' app.log (ignore case sensitive)

```
grep -n 'java' app.log (print lines along with line num)
```

```
grep -v 'apache' app.log (print lines which don't have apache keyword)
```

```
=====
Text Editors in Linux
=====
```

=> vi (visual editor) it is default editor in linux machines

=> using 'vi' we can create new files and we can edit existing files also

```
$ vi f1.txt
```

=> vi command is having 3 modes

a) command mode (just to open the file)

b) insert mode (to edit the file) ---> press 'i' in keyboard

c) esc mode (to comeout from insert mode) --> press 'esc' in keyboard

Save changes & close the file => :wq or :wq!

Without saving changes close the file => :q!

Note: vi command will open the file if it is already available otherwise it will create new file and it will open that file.

```
=====
File creation commands in linux
=====
```

touch : Create empty file

cat : Create file with data

cp : copy one file data into another file

vi : create and open file for editing

```
=====
Reading file data commands in linux
=====
```

cat : print file data from top to bottom

tac : print file data from bottom to top

rev : print each line in reverse order

head : print top 10 lines of file

tail : print last 10 lines of file

vi : open file

```
=====
SED command
=====
```

=> SED stands for stream editor

=> It is used to process file data

=> Using SED command we can perform operations on the file without opening the file.

```
# Replace first occurrence of linux keyword with unix in every line
sed 's/linux/unix/' linux-os.txt
```

```
# Replace second occurrence of 'linux' with 'unix' in every line
sed 's/linux/unix/2' linux-os.txt
```

```
# Replace all occurrences
sed 's/linux/unix/g' linux-os.txt
```

```
# Substitute and save changes in original file
sed -i 's/linux/unix/g' linux-os.txt
```

```
# delete second line of data in the file
sed -i '2d' linux-os.txt
```

```
# delete fourth line of data in the file
sed -i '4d' linux-os.txt
```

```
# delete last line of data in the file
sed -i '$d' linux-os.txt
```

```
# delete from nth line to last line (n is a number)
sed -i 'n,$d' linux-os.txt
```

```
# delete from 2nd line to 10th line
sed -i '2,10d' linux-os.txt
```

```
# print data from 3rd line to 6th line
$ sed -n '3,6p' linux-os.txt
```

```
# insert data before 4th line
$ sed '4i\i am from ashokit' linux-os.txt
```

```
# Add given text after last line
$ sed '$a\i love linux' linux-os.txt
```

```
=====
Working with User Accounts
=====
```

=> Linux is a multi user based OS When another client will connect to the same linux machine using public IP then one account will be created.

=> Multiple users can access single linux machine and can perform multi tasking

Note: "ec2-user" is default user in amazon linux vm

Note: ec-user having sudo privileges. Sudo means super user doing the operation. It acts as run as administrator in Linux

Note: For every user we can create new account to access linux vm.

```
# create user
sudo useradd <uname>
```

```
# set password for user account
sudo passwd <uname>
```

```
# display all users available in linux vm
cat /etc/passwd
# ec2-user is having sudo privilege
```

```
# switch user
$ sudo su <uname>
```

```
# Go to logged in user home directory
$ cd ~
```

```
# Delete user account
$ sudo userdel <uname>
```

```
# Delete user account along with user home directory
$ sudo userdel <uname> --remove
```

```
# how to change username
$ sudo usermod -l <new-name> <old-name>
# ec2-user having sudo privilege.
```

```
=====
What is sudoers file in Linux
=====
```

=> It is very important configuration file in linux machine.

=> Using this file we can control which user can run command as a superuser.

```
# print sudoersfile content
$ sudo cat /etc/sudeors
```

Note: We should be very careful while working with sudoers file. If we do any mistakes in sudoers file then system will be crashed.

Giving sudo previliges for user

```
# open suderos file
$ sudo visudo
```

```
# configure user like below in sudeors file (after root user details)
<username> ALL=(ALL:ALL) ALL
```

=> After making changes to close sudoers file => (CTRL + X + Y + Enter)

```
=====
How to enable password based authentication in linux ?
=====
```

=> in sshd_config file , by default PassswordBasedAuthentication is no.

=> To enable password based authentication we need to set the value as yes.

```
# Display sshd_config file data
$ sudo cat /etc/ssh/sshd_config
```

```
# Open file and enter into insert mode (press 'i')
$ sudo vi /etc/ssh/sshd_config
```

```
# set PassswordBasedAuthentication as yes
```

```
# save and close the file ( esc + :wq)
```

```
# restart sshd service
# sudo systemctl restart sshd
```

```
=====
Log Server Details
=====
```

Public IP : 43.204.143.144
username : loguser
pwd: log@123

use below command to connect with username and pwd
\$ ssh uname@public-ip

Note: To connect with linux vm using username and password then passwordbasedauthentication must be enabled in that linux vm.

=====
Working with User Groups
=====

=> When we create user in linux, for every user one user group also will be created with the given username.

display all groups available
\$ cat /etc/group

create new group
\$ sudo groupadd <group-name>

Add user to group
\$ sudo usermod -aG <group-name> <user-name>

delete user from group
\$ sudo gpasswd -d <username> <group-name>

=====

- 1) File permissions & ownership
 chmod & chown
- 2) Networking commands
- 3) Package Managers (s/w installation)
- 4) Linux Architecture

=====
File Permissions
=====

ec2-user => /home/ec2-user ==> f1.txt

raju => /home/raju ==> f1.txt

rani => /home/rani ==> f2.txt

ashok => /home/ashok ==> f3.txt

=> We can create several user accounts in single linux vm

=> Multiple users can connect to single linux vm at a time.

Note: One user can modify the file created by other user in linux vm.

=> To avoid this problem we will use file permissions in linux

=> In Linux, file permissions are divided into 3 types

r => read

w => write
x => execute

=> A file/directory contains 3 sections of permissions

user (owner) => u

group => g

others => o

=> We can see below permissions for a file & directory

rw-rw-rw- f1.txt

user having read + write + execute
group having read + write + execute
others having read + write + execute

rw-r--r-- f2.txt

user having read & write (no execute permission)
group having only read
others having only read

rw-r-xr-x java

user having read + write + execute
group having read + execute
others having read + execute

r-xr----- sbms

user having read + execute
group having only read
others having only execute

=> To change file permissions we will use 'chmod' command

Giving execute permission for user
\$ chmod u+x f1.txt

giving write permission for group
\$ chmod g+w f1.txt

Remove execute permission for others
\$ chmod o-x f1.txt

Remove all permissions for others
\$ chmod o-rwx f1.txt

give all permissions for group
\$ chmod g+rwx f1.txt

=====
File Permissions in Numeric Format
=====

0 => No Permission
1 => Execute
2 => Write
3 => (2+1) => Write + Execute
4 => Read

```
5 => (4+1) => Read + Execute
6 => (4+2) => Read + Write
7 => (4+2+1) => Read + Write + Execute
```

```
$ chmod 765 f1.txt
```

- user having all permissions
- group having read + write
- others having read + execute

```
$ chmod 456 f1.txt
```

- user having only read
- group having read + execute
- others having read + write

```
=====
Ownership change
=====
```

=> To change file/directory ownership we will use 'chown' command

```
# changing owner
sudo chown new-owner file/directory
```

```
# changing owner-group
sudo chown :new-group file/directory
```

```
# changing owner & group
sudo chown new-owner:new-group file/directory
```

```
=====
Q) What is the diff between chmod & chown ?
=====
```

chmod => To change file/directory permissions

chown => To change owner/group

```
=====
Networking Commands
=====
```

ping : To check connectivity

```
$ ping www.google.com
$ ping www.google.com
$ ping 192.168.1.20
```

wget : It is used to download the files from internet

```
$ wget https://d1cdn.apache.org/tomcat/tomcat-9/v9.0.91/bin/apache-tomcat-9.0.91.zip
```

curl : To send http request to server (api call)

```
$ curl https://type.fit/api/quotes
```

ifconfig: To get IP address of our machine

```
$ ifconfig
```

```
=====
```

```
whoami
pwd
date
cal
cal 2050
mkdir
rmdir
touch
ls -ltr
cat
cp
rm -rf
mv
tac
head
tail
grep
vi
sed
```

```
useradd
userdel
usermod
groupadd
groupdel
id
```

```
chmod
chown
```

```
ping
wget
curl
ifconfig
```

```
=====
Package Managers in Linux
=====
```

=> Package Managers are used to install softwares in linux machines

=> Package Managers are specific to linux distribution

Amazon Linux + Red HAT : yum

Ubuntu Linux + Debian : apt

```
#install git client s/w
sudo yum install git -y
```

```
# install java s/w
sudo yum install java
```

```
# install maven s/w
sudo yum install maven
```

```
=====
Install WebServer in Linux VM
=====
```

=> Webserver is a software which is used to run websites

=> Websites are divided into 2 types

1) static website (fixed content)

ex : wikipedia

2) dyanmic website (content will change based on user)

ex: gmail, facebook

=> For static websites execution we can use 'httpd' as webserver

=> For dynamic websites execution we can use 'tomcat' as webserver

```
# install httpd webserver
sudo yum install httpd -y
```

```
# start httpd server
sudo service httpd start
```

Note: httpd webserver runs on http protocol which is 80.

To access our webserver we need to enable 80 port number in EC2 VM security group inbound rules (firewell setting)

=> Access our webserver using EC2 VM public ip address in our browser.

```
# Navigate to webserver directory
cd /var/www/html
```

```
# create index.html file with content
sudo vi index.html
```

```
=====
What is systemctl in linux ?
=====
```

=> systemctl is a command-line utility in Linux systems which is used to manage system services

```
=> Starting service
=> stopping service
=> restarting service
=> reloading service
=> enabling / disabling services
```

```
#check service status
sudo systemctl status <service_name>
```

```
#start service
sudo systemctl start <service_name>
```

```
#stop service
sudo systemctl stop <service_name>
```

```
#re-start service
sudo systemctl restart <service_name>
```

```
=====
How to change hostname in linux vm ?
=====
```

```
# set hostname
```

```
$ sudo hostname <new-name>
```

```
# re-start session
```

```
$ exit
```

Note: Connect back to linux vm using ssh command.

```
=====
whereis command
=====
```

To know the location of the package we have installed

```
whereis java
```

```
whereis maven
```

```
whereis git
```

```
=====
find command
=====
```

=> find command is used to search files location

```
# find the file whose name is oops.txt
```

```
sudo find /home -name oops.txt
```

```
# find all the empty files inside /home
```

```
sudo find /home -type f -empty
```

```
# find all the empty directories inside /home
```

```
sudo find /home -type d -empty
```

```
# find the files which are 30 days old in linux vm
```

```
sudo find /home -mtime 30 -print
```

```
=====
Assignment
=====
```

Deploy Spring Boot Application in Linux VM : <https://www.youtube.com/watch?v=cRQPgbwOWq0>

```
=====
Summary
=====
```

1) Why Linux for Java developers ?

2) What is OS & why ?

3) Windows Vs Linux

4) Linux History

5) Linux Distributions

6) Linux VM Setup

7) Linux commands

8) Working with directories & files

9) Working with editors (vi & sed)

- 10) Users & groups Management
- 11) sudoers file & sshd_config file
- 12) PasswordBasedAuthentication enable
- 13) File Permissions & Owership
- 14) Package Managers
- 15) Services management (systemctl)