SESSION-01 **(Cover entire video)**

- Small STORY
- The SDLC
- **Waterfall vs agile vs DevOps**
- What is DevOps?
- Why Linux?

SMALL STORY:-

SMS ➔ School Management System

Stakeholders in SMS are Teachers, Parents, and Students

100 years back → Final exam

1. Teachers → Not serious to **complete syllabus** from DAY-1
2. Parents → Yes. Worried about, whether they **pass or not** from Day-1
3. Students → Not serious to **study** from DAY-1

Final pass percentage at the end is 30%

Process Changed

UNIT TEST-I, II, III, IV

Q, H, Pre-Final, Final

UNIT TEST-I → 30 days

Teachers → They should be **serious from DAY-1 to complete syllabus** for UNIT TEST-I

Students → One week before they start reading for UNIT TEST-I

Parents → They are waiting for UNIT TEST-I results

Final pass percentage at the end is 80%

Software Development Life cycle (SDLC):-

1. Requirements analysis
2. Planning (or) feasibility study
3. Design → **General requirements** to **technical requirements**
4. Implementation
5. Testing
6. Deployment
7. Maintenance

Parents → Clients

Teachers → IT Management

Students → Developers, Operations team, testing team

Waterfall:-

1. Requirements analysis
2. Planning
3. Design → General requirements to technical requirements
4. Implementation
5. Testing
6. Deployment
7. Maintenance

Requirements → Phase-I

Once you are in Phase-II, You can't go back and change the requirements.

6 months for development → Testing and Deployment

100 defects → 10 invalid defects

50 defects → 6 invalid defects

Ambassador (we delivered Ambassador but client's expecting BMW)

Agile:-

1. Requirements analysis
2. Planning
3. Design → General requirements to technical requirements
4. Implementation
5. Testing
6. Deployment
7. Maintenance

Modules → Signup, Login, Menu, Order, Shipping, Delivery, Payments

Signup → 1 month

2 weeks for development, 2 weeks for testing and deployment → it is called sprint

Development is serious from DAY-1

20 defects → 5 invalid defects

Honda City (we delivered Honda City but client's expecting BMW)

- Waterfall → 10 times testing → 100(bugs are identified in testing)
- Agile → 30 times testing → 101 (bugs are identified in testing)
- DevOps → 100 times testing → 102

Agile is a part of DevOps:-

Modules → Signup, Login, Menu, Order, Shipping, Delivery, Payments.

Signup → 1 month

DAY-1 → 100 lines of code → enter your first name, enter your last name

Deploy this 100 lines → test this 100 lines

DAY-2 →

---- And so on till Day-9------

DAY-10 → we should deploy and test everything.

DevOps is a process of building, testing and releasing code on the same day when developer writes something.

- **Through this process we can achieve Co-operation, Co-ordination, and Collaboration between teams.**
- **Faster releases and**
- **Less defects.**
- **Speed and Accuracy**

In different organisations or projects having different environments like.

DEV, PROD

DEV, QA, PROD

DEV, QA, UAT, PRE-PROD, PROD

DEV, QA, UAT, PERF, SECURITY, PRE-PROD, PROD

Linux (It is a case sensitive):-

Windows VS Linux:-

We need to restart sometimes → It can run for years

Too many graphics → Time to load → No graphics → Super performance

More Resource consumption → less resource consumption

Not that much secure → Secure

Costly → Free

Not open source → Open source

Tomorrow we will discusses below topics.

Server Creation, Linux Commands, Editors, User creation, Install software's, Service management.

AWS Account creation https://youtu.be/F4jF88UkxV4 -- Video link to create AWS account.

Basic laptop specifications to practice this course:

i3, 8GB ram

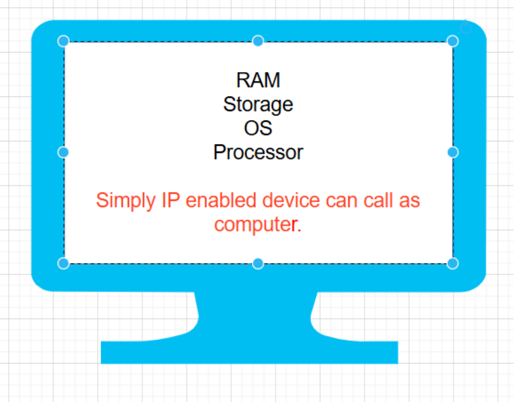
i5, 16GB

i7, 16GB

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

Session-2 (just focus till 1hour 17mints)

What is a Computer?
Client-Server Architecture
Authentication mechanisms
Public key and Private Key
Firewall creation **@16 mints**
Import Key pair
Instance Creation

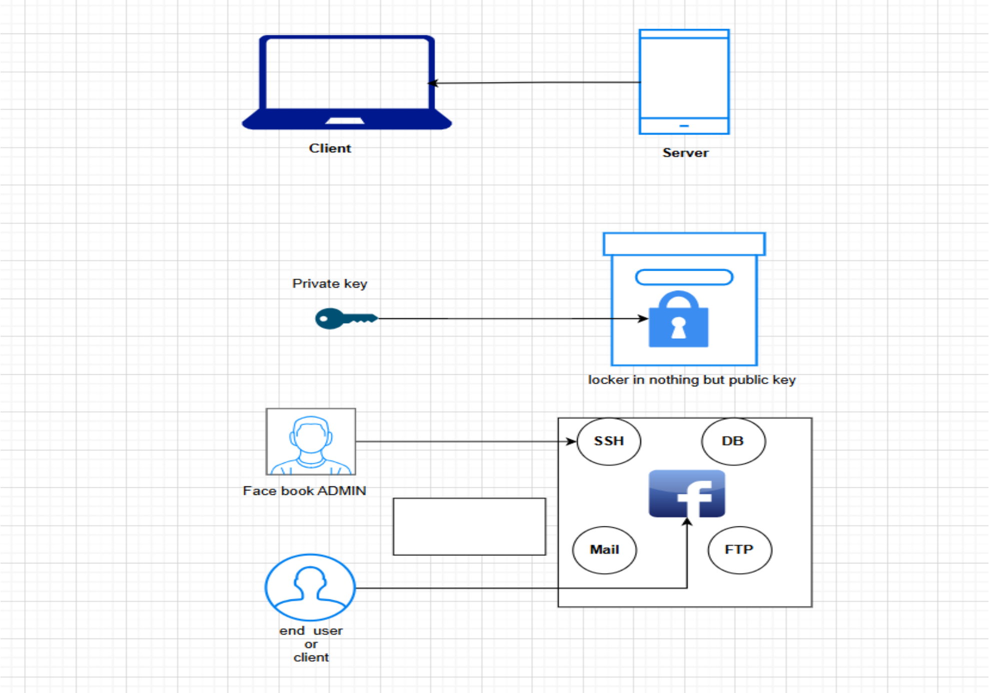


- **Server → The device which is used to host an application**

Port forwarding → you can deploy application in your laptop and open it in internet (we can see this option in modems)

Facebook.com → Facebook application in fb servers

Browser → Client software



For any server companies (like AWS, Azure) below things are primary to set their business.

- **Region** → HYD, Mumbai, Singapore, US, EU
- **Available Zone (AZ)** → North HYD, South HYD → minimum 2 AZ → High availability

Instance OR **Server** OR **node (**all are same only but different words**).**

LAUNCHING OF EC2 INSTANCE ON AWS PLATFORM:-

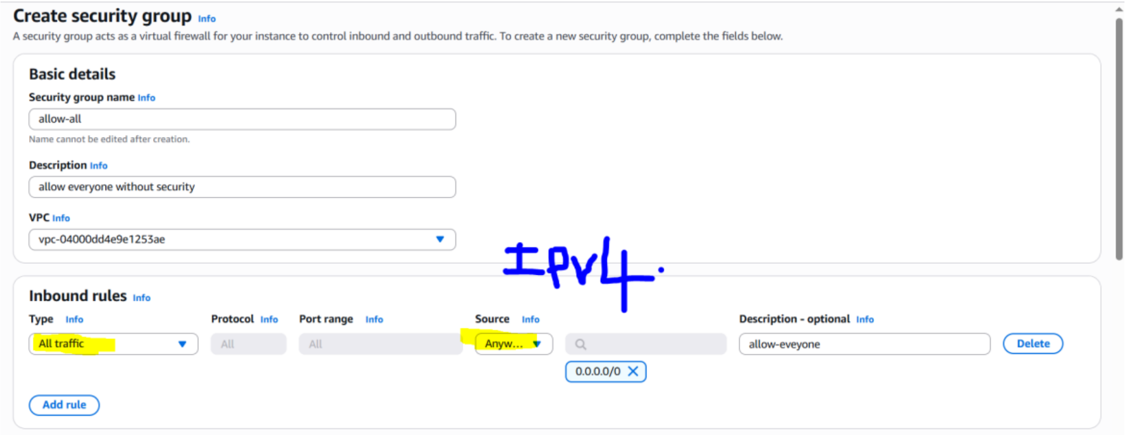
Firewall OR **Security Group (**all are same only but different words**).**

Allow everyone one through firewall

Inbound → incoming traffic

Outbound → outgoing traffic

0.0.0.0/0 → means every computer in the internet



Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info: allow-all

Name cannot be edited after creation.

Description Info: allow everyone without security

VPC Info: vpc-04000dd4e9e1253ae

IPv4.

Inbound rules Info

Type	Protocol	Port range	Source		Description - optional	
All traffic	All	All	Anyw...	0.0.0.0/0	allow-eveyone	Delete

AMI (Amazon Machine Image) → devops-practice

Red Hat Enterprise Linux (RHEL) == Centos == Amazon Enterprise Linux == Fedora == Alma Linux

.iso → The International Organization for Standardization

1. ISO stands for **The International Organization for Standardization.**
2. which is nearly followed by 91 nations(countries)
3. Here **.iso** is a type of file extension.
   It will create entire operating system → C:\Windows

Previously back in 1980s.

Mac → Hardware locking → you are buying both hardware and Mac OS

Servers == UNIX

Laptops == IBM BIOS (BIOS stands for Basic Input/Output System.) The Epson HX-20 was the first laptop.

Linus Torvalds is the Inventor of Linux

OS (operating system) == Kernel + User Interface

Kernel == Brain of OS == C language is used develop kernel of Linux.

Linux (Kernel) → C language → Open Source

Linux Implementations or **Distributions** or **Flavours of Linux:-**

RHEL → Commercial (Enterprise)

IBM AIX

Ubuntu → most popular open source OS**.**

Fedora

Solaris

Suse

Android

Open Source Vs Enterprise

Here's a clear comparison of **Open Source** vs **Enterprise** software:

Feature	Open Source	Enterprise (Proprietary)
Source Code	Available and modifiable by anyone	Closed and owned by the company
Cost	Usually free or low-cost	Typically requires a paid license or subscription
Customization	High – can be modified as needed	Limited – changes need vendor support
Support	Community-driven (forums, contributors)	Professional support from the vendor
Security	Transparent but may need expert management	Managed by vendor; security patches are provided
Updates	Frequent, by the community	Regular, controlled updates from the vendor
Ease of Use	May require technical skills	User-friendly, designed for businesses
Examples	Linux, Apache, MySQL, LibreOffice	Windows Server, Oracle DB, Microsoft Office 365

Summary:

- **Open Source** is ideal for flexibility, control, and cost-saving.
- **Enterprise Software** is better for businesses needing professional support, warranties, and ease of use.

AWS Linux 2023 AMI

t3.micro/t2.micro → these are the free tire options for launching EC2 instance.

Authentication:-

1. What **you know** ➔ Username and Password
2. What **you have** ➔ Username and token/OTP
3. What **you are** ➔ Fingerprints, Retina, Palm, etc.

Public Key and Private Key

Lock and Key

Lock ➔ Public (.pub extension file)

Key ➔ Private (.pem extension file)

Git bash → SSH Client, Git Client, Mini Linux

~ ➔ Home directory

pwd ➔ Present working directory

C:\Users\<your-username> ➔**Windows format of paths**

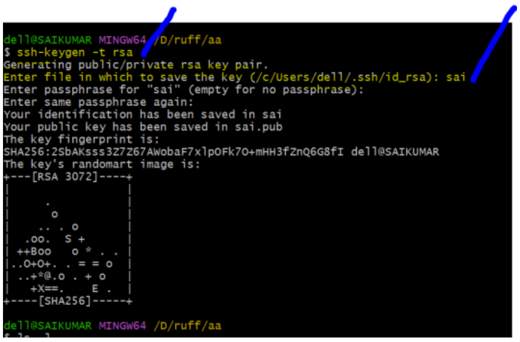
/c/Users/sivasai ➔**Linux format of paths**

ls -l ➔ List sub directory in current path

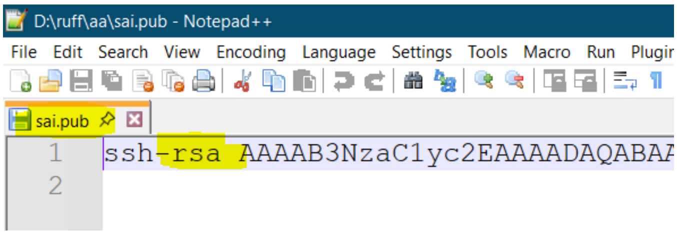
ssh-keygen -f <file-name> ➔ this command line will gives us public key and private key

ssh-keygen –f sai → this generates two files sai, sai.pub.

ssh-rsa



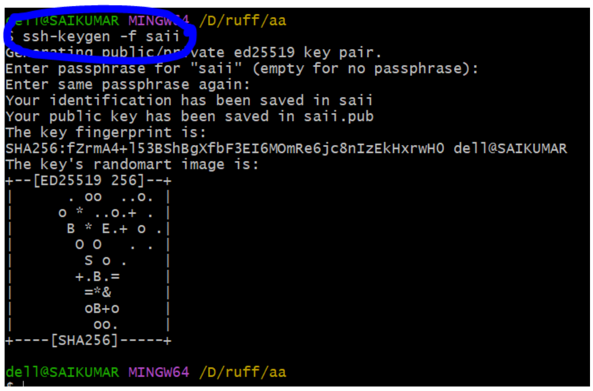
```
dell@SAIKUMAR MINGW64 /D/ruff/aa
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dell/.ssh/id_rsa): sai
Enter passphrase for "sai" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in sai
Your public key has been saved in sai.pub
The key fingerprint is:
SHA256:2SbAKsss3Z7Z67AWobaF7xlpOFk7O+mHH3fZnQ6G8fI dell@SAIKUMAR
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|      .          |
|       o         |
|     .. . o      |
|   .oo.  S +     |
|  ++Boo   o * . .|
|..O+O+. . = = o  |
| ..+*@.o . + o   |
|   +X==.    E .  |
+----[SHA256]-----+

dell@SAIKUMAR MINGW64 /D/ruff/aa
```



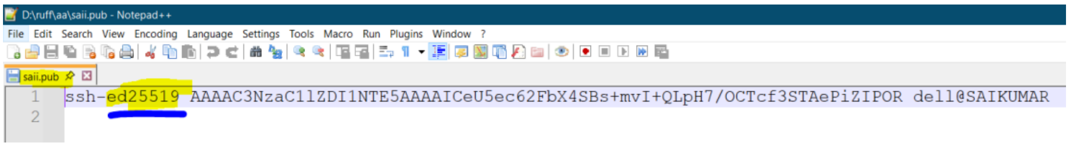
D:\ruff\aa\sai.pub - Notepad++

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAA
```

ssh-ed25519



```
dell@SAIKUMAR MINGW64 /D/ruff/aa
$ ssh-keygen -f saii
Generating public/private ed25519 key pair.
Enter passphrase for "saii" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in saii
Your public key has been saved in saii.pub
The key fingerprint is:
SHA256:fZrmA4+153BShBgXfbF3EI6MOmRe6jc8nIzEkHxrwH0 dell@SAIKUMAR
The key's randomart image is:
+--[ED25519 256]--+
|       . oo  ..o.|
|     o * ..o.+ . |
|      B * E.+ o .|
|       o o    . .|
|        S o .    |
|       +.B.=     |
|        =*&      |
|        oB+o     |
|         oo.     |
+----[SHA256]-----+

dell@SAIKUMAR MINGW64 /D/ruff/aa
```



D:\ruff\aa\saii.pub - Notepad++

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICeU5ec62FbX4SBs+mvI+QLpH7/OCTcf3STAePiZIPOR dell@SAIKUMAR
```

Any one of above words (**ssh-ed25519 or ssh-rsa**) can be a part of **public keys.**

Enable extension

Control Panel\All Control Panel Items

File Explore options

View

Uncheck Hide extensions for known file type.

jul-payslip.pdf

jul-payslip

.pub extension refers to public key

.pem extension refers to private key

Public IP == 34.227.191.107 (we have to access the server, with public IP address only).

AWS Linux 2023 AMI → **ec2-user** is default user ID and our **private key** is the passwords for logging into the server

IP, Username, password, **protocol**, port

HTTP facebook.com/IP, Username, Password**, 80**(it is default port number for HTTP)

Similarly for HTTPS-----**443** (it is default port number for HTTPS)

Facebook.com/IP, Username, Password/Private Key, SSH, 22

Secure Shell (SSH) --> it will give full access to the server and default port number for this protocol is 22.

Delhi --> HYD

f:no, apartment name, pin code

siva, 523764

```
ssh -i <private-key> ec2-user@IP
```

Task to complete after the class:-

1. Create public and private keys
2. Import public key
3. Create firewall
4. Create Instance
5. Connect to Instance
6. Terminate when not using.

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

Session-3 (just focus till 1 hour 10mints)

- **Linux basic commands**

uname ➔ gives us kernel name

echo "Hello world" ➔ this is preferable to print on the screen itself

echo 'Hello world' ➔ this command also can print but not preferable

ssh -i <path-to-your-private-key> ec2-user@IP➔command to login to the server from **git-bash by using private key**

absolute path and relative path

/c/devops/daws-82s/daws-82s.pem ➔**absolute path**

ssh -i /D/Devops/DAWS-82S/Key_Pairs/sai_test.pem ec2-user@18.207.125.78

daws-82s/daws-82s.pem ➔ **relative path**

cd /D/Devops/DAWS-82S/Key_Pairs/

ssh -i sai_test.pem ec2-user@18.207.125.78

clear ➔ clears the current console

$ ➔denotes normal user

sudo su ➔root access

# ➔ denotes admin/root user

/home/<user-name> ➔ normal user home folder

sudo su - ➔lands into **root user's home folder**. (/root)

/root ➔ root user home folder (type **sudo su** enter. If we type CD then enter we will land in to root's home directory)

cd / ➔ we will land in to root folder of the server (it is like THIS PC of the windows)

<command-name> <options> <inputs>

- ➔ We can give single char

-- ➔ We need to give word

ls ➔ list subdirectories

ls -l ➔long listing format with more details

ls -lr ➔reverse alpha order

ls -lt ➔new files on top

ls -ltr ➔old files on top

ls -ltrh ➔human readable

ls -la ➔display all files including hidden files and folders

drwx------ ➔**d** means directory

-rw-r--r-- ➔ **-** means file

lrw-r--r-- ➔ **l** means Link files

touch <file-name> ➔ creates empty file(any file we can create with it but take care of file extensions)

cat devops.txt ➔ to open or view the file

cat > devops.txt ➔ enter the text, once done **enter** and then press **ctrl+d**

>> ➔ append, adds to the current text

Example: cat >> devops.txt

rm devops.txt ➔ to remove the file from server in relative path

rm -rf devops.txt ➔ for force delete of file

mkdir <name> ➔ creates directory

rmdir <name> ➔ **removes only empty directory**

rm -r devops ➔ ***recursively delete everything inside devops folder***

cd .. ➔ One step back from current folder or directory

=======================at 32mints ====================

cp <source-file> <destination> ➔ we can use this command when we are in FROM folder and we are PUSING THE FILE.

cp passwd /home/ec2-user--------(I performed it from **/etc** folder)

=======================at 46mints ====================

cp <source file along with path> <**save as** the file in your current path> ➔ coping file with name we want PULLLING THE FILE PROCESS

cp /etc/passwd passwd--------(I performed it from /home/ec2-user)

cp /etc/passwd passwddddd

curl and wget

wget <URL> ➔ Downloads the file

curl <URL> ➔ Shows the content on the screen

curl <URL> **-o <save as>** ➔ it will downloads the content and saves the file as the name we provided

Raw text URL from git:-

https://raw.githubusercontent.com/DAWS-82S/notes/refs/heads/main/session-02.md

In above url / ➔ **Separator** or **delimiter** or **fragments**

Sivakumar Reddy M

In above name, space is the separator or delimiter

At 46mints sir explained HOW TO CHECK NUMBER OF USERS CONNECTED TO THIS SERVER, (/etc/passwd → we should **only check** in this file but don't corrupt the file).

grep <word-to-search> <file>

grep ec2-user passwd

Piping

| ➔ pipe

cat <file-name> | grep <word-to-search>

cat passwd | grep ec2-user

Cut command:-

Command: echo "https://raw.githubusercontent.com/DAWS-82S/notes/refs/heads/main/session-02.md" | cut -d "/" -f1

Output: https:

Command: echo "https://raw.githubusercontent.com/DAWS-82S/notes/refs/heads/main/session-02.md" | cut -d "/" -f9

Output: session-02.md

awk command: -

Command: echo "https://raw.githubusercontent.com/DAWS-82S/notes/refs/heads/main/session-02.md" | awk -F "/" '{print $NF}'

Output: session-02.md (NF always gives us last fragment).

Command: echo "https://raw.githubusercontent.com/DAWS-82S/notes/refs/heads/main/session-02.md" | awk -F "/" '{print $1F}'

Output: https: (1F always gives us last fragment).

How can I get all the users in Linux Server?

awk -F ":" '{print $1F}' passwd ➔ for total user names

awk -F ":" '{print $3F}' passwd ➔ for total user Ids.

man <any key word you want to search> ➔ this command is like manual book of key words

q ➔ Quit from manual mode

head passwd ➔ top 10 lines from top

tail passwd ➔ last 10 lines from Bottom

tail -n 4 passwd ➔ last 4 lines from bottom

head -n 3 passwd ➔ top 3 lines from top

head -n 10 passwd | tail -n 7 ➔ this query displays lines from line 4 to 10

head –n 10 fileName | tail (10-4)+1======some kind of formula=======

Since we are searching in root folder we must need root access then only it will work

find <which-location> -name "<file-name>"

find / -name "passwd" ➔ this command gives us all files, all directories with this name.

find / -type f -name "passwd" ➔ this command gives us only files with this name.

find / -type d -name "passwd" ➔ this command gives us only directories with this name.

or

find <which-location> -type f "<File name>"

find / -type f -name "**mm**"

or

find <which-location> -type d "<Directory name>"

find / -type d -name "Amma" ➔ this command gives us all directories, with this name.

find / -type d -name "**mm**"

vim editor(visually improved editor) ➔ will be discussed on next session.

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

Session-4 (just focus till 1 hour 24 mints)

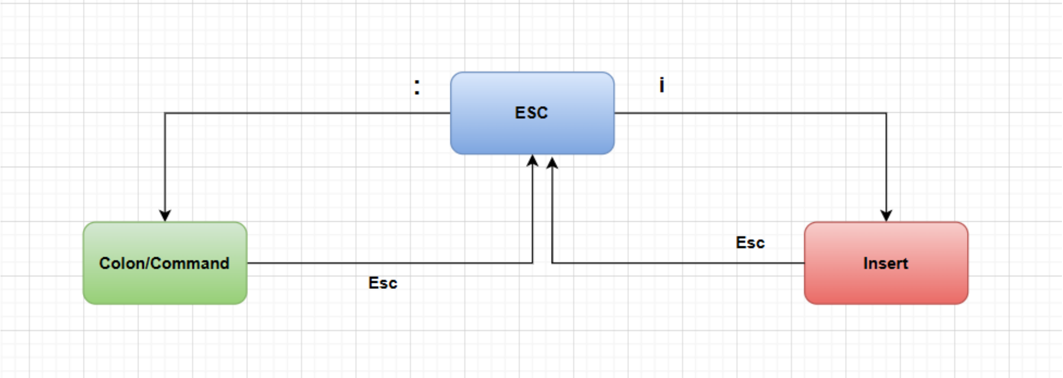
- **vim editor**
- **user management**

cp /etc/passwd passwd

vim passwd → vim command is used to open the file in vim editor.

- ESC mode
- : Mode (or) Command mode----- (colon or command mode)
- Insert mode

- **Esc mode is default one**



Press : to enter into command mode

COMMAND MODE

:q → quit

:wq → write and quit

:q! → Force quit without changes

:wq! → Means “save the file and quit, even if it’s read-only or normally restricted.”

:set nu → set line numbers in the file

:set nonu → don’t set line numbers

:/<word-to-find> → search the word from **top to bottom** and word is highlighted.

:?<word-to-find> → search the word from **bottom to top** and word is highlighted.

:noh → no highlight

:28 d → deletes 28th line

:3s/word-to-find/word-to-replace → replaces first occurrence in that(3rd) line

:3s/word-to-find/word-to-replace/g → replaces all occurrences in that line

:%s/word-to-find/word-to-replace/g → replaces all occurrences in file

:%d → delete entire content

ESC Mode

U ➔ undo

yy ➔ copy the line

p ➔ paste

10p ➔ paste the line 10 times

dd ➔ cut the line

gg ➔ takes cursor to top

shift+g ➔ takes cursor to bottom

Insert Mode

press i

SUDO ACCESS NEED, TO PERFROM ADMIN RELATED ACTIVITIES

Linux Administration

- User management

User management

Create user, add user to any group

useradd <user-name> ➔ creates user and group with same name

id <user-name> ➔ displays user information (admin credentials not required)

cat /etc/passwd ➔ contains user information (admin credentials not required)

cat /etc/group ➔ contains group information (admin credentials not required)

In Linux, a user must have only one primary group and at least one secondary group

passwd <user-name> ➔ sets password to the user

groupadd <group-name> ➔ creates group

usermod -g devops ramesh ➔ adds ramesh to devops group (here –g refers to primary group)

usermod -aG testing ramesh ➔ adds testing as secondary group (or) we can add a user to n number of groups with this command. (here –aG "G" refers to secondary group. "-a" refers to append)

CRUD ➔ means Create Read Update Delete (here Siva explained an incident which is occurred at Singapore …about hacking)

userdel <user-name>

First remove from primary project or projects (we have to reassign the person to default user group which is created by default while creating user id for the first time), then remove ID from company.

{usermod –g <user name> <user name> ➔ then userdel <user-name> }

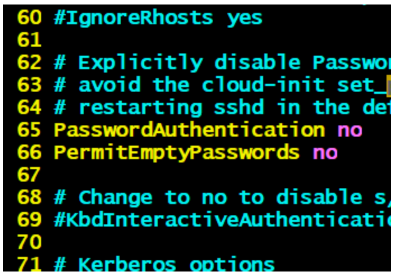
groupdel <group name> ➔ that particular group will be deleted

getent group <group name> ➔ that particular group details will be displayed

Linux follows key based authentication by default

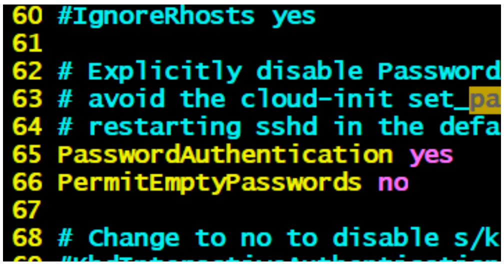
vim /etc/ssh/sshd_config ➔ edit the SSH related configuration

> /etc/ssh/sshd_config ➔ any mistakes in this file, we can't get in to the server through **ssh**. If it happened then we have to factorial reboot the server.



```
#IgnoreRhosts yes

# Explicitly disable Passwo
# avoid the cloud-init set_
# restarting sshd in the de
PasswordAuthentication no
PermitEmptyPasswords no

# Change to no to disable s
#KbdInteractiveAuthenticati

# Kerberos options
```

--------------->



```
#IgnoreRhosts yes

# Explicitly disable Password
# avoid the cloud-init set_pa
# restarting sshd in the defa
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/k
```

systemctl restart sshd

sshd -t ➔ to check config is correct or not(if any issue we will get error after running this command)

key based authentication ➔ Ramesh should generate his public key and private key…it is covered in next class

==================My Practice part===================

:/<word to find in the text> hit enter, to navigate among or between lines hit n.

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

Session-5 (Cover entire video)

- User management
- Permissions and Ownership
- Package management
- Service management
- Network management

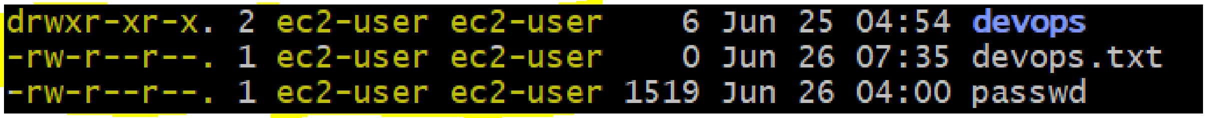
USER MANAGEMENT (mostly Linux admin team works on this)

Permissions

r == 4 ➔➔ R indicates **read** permission

W == 2 ➔➔ W indicates **write** permission

X == 1 ➔➔ X indicates **Execute** permission



```
drwxr-xr-x. 2 ec2-user ec2-user    6 Jun 25 04:54 devops
-rw-r--r--. 1 ec2-user ec2-user    0 Jun 26 07:35 devops.txt
-rw-r--r--. 1 ec2-user ec2-user 1519 Jun 26 04:00 passwd
```

-rw-r--r--. 1 ec2-user ec2-user 0 Feb 4 07:28 practice

- rw- r-- r-- **ec2-user ec2-user**

file |<owner-who-created> <group> <others>

ec2-user → Owner → Read and Write

ec2-user → Group → Only Read

Others → Other than owner and group → only read

Who can change permissions of file **or** folder?

Ans: owners **or** root user

Owner/user **==** u

Group **==** g

Others **==** o

chmod <privileges to grant or remove for u, g, o> <file or folder name>

chmod o+w devsecops.txt ➔ This command gives write access to others

chmod o-r suresh.txt ➔ This command removes read access to others

chmod ugo+rwx suresh.txt

chmod 740 suresh.txt

chmod 777 suresh.txt

chmod 755 suresh.txt

Pub key Authentication for newly added users to the server:-

Admin should ask for saii public key

With root access follow below steps

```
cd /home/saii/
mkdir .ssh
cd .ssh
touch authorized_keys
```

vim authorized_keys → place the saii's public key properly and save the file, make sure that no extra spaces.

```
chown saii:saii authorized_keys
chmod 400 authorized_keys
```

chown -R saii:saii .ssh → here -R means regressive.

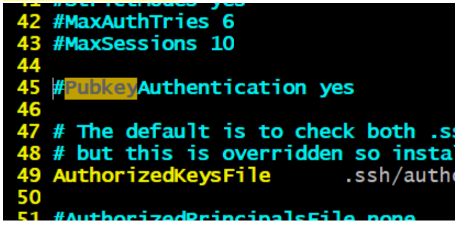
chmod 700 -R .ssh/ → **Apply the change recursively — to the directory and all files/subdirectories inside**

```
ls –l
cd
```

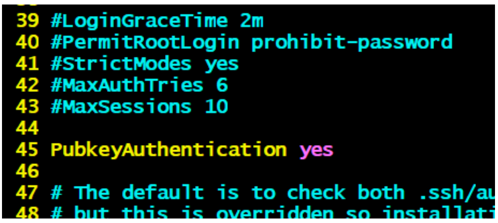
then some configurations needs to change

```
vim /etc/ssh/sshd_config
```

uncomment PubkeyAuthentication yes and save it



```
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .s
# but this is overridden so insta
AuthorizedKeysFile      .ssh/auth

#AuthorizedPrincipalsFile none
```



```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/au
# but this is overridden so installati
```

```
sshd -t
systemctl restart sshd
```

```
ssh -i saii.pem saii@IP
```

Package management

Windows laptops are configured with URL to pull the updates.

Usually one package depends on other packages...

/etc/yum.repos.d/ ➔ here we have all repos URL, which are used to install packages when we need.

Yum and **dnf** commands are part of RHEL

yum install <package-name>

dnf install <package-name> → apt-get is for ubuntu

apt install git

dnf remove <package-name>

dnf update <package-name>

sudo su -

dnf **install** git –y

dnf install gcc -y

dnf **remove** git –y

dnf remove gcc -y

dnf **update** git -y

dnf list installed ➔ displays all already installed inside Linux

dnf list installed | wc -l ➔ it gives us count value of number of packages installed in server.

dnf list available ➔ all – installed(all minus installed means what are the extra apps that we can install)

Service Management

ssh -i <private-key> suresh@IP

request goes to IP, checks SSH is running on port number 22...

systemctl status sshd ➔ d for demon and to login to server, sshd should be ALWAYS in running state.

http service or server, for http service nginx or apache (nginx is latest one apache is old trend)

dnf install nginx -y ➔ to install packages with only one instruction or query.

systemctl start nginx ➔ to start nginx service

systemctl status nginx ➔ to check status

http://IP:80 ----- ekkada port number 80 physical ga mana ki kanpinchadu.

systemctl stop nginx

systemctl restart nginx ➔ restart

systemctl enable nginx ➔ services will start automatically

systemctl disable nginx

Few packages are just utilities, they are command line packages.

Example: systemctl start git ➔ invalid

Few packages are service related, we can start/stop/restart/enable.

systemctl start nginx

systemctl status nginx

Process Management

TL

Senior

Junior

Fresher

TL → TASK-1 → Senior

Senior → TASK-2 → Junior

Junior → TASK-3 → Fresher

For TASK-3, TASK-2 is the parent.

For TASK-2, TASK-1 is the parent.

Process ➔ in linux everything is process

echo "Hello World" ➔ creates one process instance id

gives the result and then mark the process as completed ps ➔ simple search of all app’s or service’s process status

ps -ef ➔ advanced search

PPID ➔ parent process instance id

```
dnf install nginx-y
systemctl start nginx
ps -ef | grep nginx
```

Foreground and background

sleep 10 & [ampersand ani palakalli it will allow that instruction(sleep) run in background…..sleep for 10 seconds]

```
sleep 100 &
ps -ef | grep sleep
```

kill PID ➔ request to stop ------It terminates that PID

kill -9 PID ➔ order to stop (here nine is mandatory one) ------It kills that PID

Network Management(it will discussed with AWS VPC) for troubleshooting

netstat –lntp ➔ to check ports (list of network ports in operation)

systemctl status nginx

ps -ef | grep nginx

netstat -lntp ➔ check port is open or not

=============== My Practice part =================

for renaming: mv filename < save as>

example: - mv passwd END

If SSH’s PID is killed all users disconnects from server and not able to connect back to fix this we have to reboot server.

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺