apache deploying location :- /var/www/html

tomcat deploying location :- webapps - service tomcatstart

.M2 is the maven Local Repo.

Service nexus status : Nexus.

Jenkins configuration file :- cd /etc/sysconfig/

cat /etc/Passwd to see users

maven dependency are under POM.xml file (Java)

Nodejs dependency are under package.json

Jenkins = 8080. cd /etc/sysconfig.

sonarqube = 9000.

Nexus = 8081

Tomcat = 8080

Apache = 80

infrastructure

Ansible ~~instr~~ managment tool

Terraform (IAC) Infrasture as a Tool.

"Hashicrop"

AWS instance creation:- 02/02/22

⇒ EC2 - Elastic cloud compute (Default user)

* instace creation is a '7' step Process

i) Choose an Amazon Machine Image (AMI)

ii) Choose an instance Type

iii) iv) v) vi) vii) leave as Default.

* Select an existing (or) create a new 'key Pair':-

i) create a new key pair (or) if existing select that one.

ii) key Pair type.

● RSA ○ ED25519

iii) key pair Name.

ex:- vamshi-devops.

⇒ Download key Pair

⇒ launch instance.

⇒ View instance.

Putty and Putty gen :-

Putty } download
Putty gen

Putty gen :- (.PPK file is needed)
{* so. to convert .Pem to .PPK we use putty *}

Load
→ Go to download
Change to All files

Select .Pem file

Save private key.

file name :- ex:- Vamshi-deveops.

Vamshi-deveops PPK file is present in windows.

* To connect

Putty :-

Host Name (IP address).

(Paste) IPv4 public IP

⊞SSH

AUTH.

Browse (select the .PPK)

open

login as : ec2-user. (Default user).

03/02/22

GIT BASH:- (NO .PPK file is needed)

* cd Downloads/ (cd - change directory)

* Paste (Path).

* Yes

* Sudo -i (root user).

*1) whoami (ec2-user). {who connected to Machine}

*2) PWd (Present working directory).

3) ls (ls-list) (To see content in particular directory)

4) ls -a (To see hidden content).

5) ls -l -a (To see if it is a file (or) directory).

6) Man ls (To seek help of commands).

7) touch hello.txt (~~vi~~ touch Editor to create files).
Empty files.

8) ls

9) ls -l (long list).

10) cat (file name). (To read the content of fiel).

11) echo "Hello All" (echo to display a message).

12) vi (file name) (vi Editor to place content in files).
< (read) > (write)
(if present it will open (or) if the file is not present it will create a new file)

12) i) Press I → insert

→ Escape ←

ii) :wq (to save).

iii) :q! (to exit without saving).

iv) yy (copy content).

v) P (paste).

vi) Number yy (multiple copy of record).

vii) dd (delete data).

viii) :/word (find).

ix) :%.s/ (find & replace).

ex:- :%.s/This/That.

13) Mkdir (name) (To create direcotory).

14) cd (name/) (To go to the directory).

ex:- cd devops/
ls.
git
cd git/
Pwd
deveops/git/

(To create Sub-directory first Go to the ~~paren~~ parent dire and then with command mkdir create another directory) the switch to the Sub-directory).

15) cd .. (To switch one-level back).

16) cd devops/git/ (To switch directly to the directory wanted [parent/child]). Forward

cd ../..   (TO go back particular Level).

cd   (will take to the home level).

17) cp (source) (destination)   (copy content b/w Locations).
ex :- cp jenkins devops/

18) *   (vild card Charracter
* = all)

19) cp -r (source) (destination)   (whenever dealing with directories use '-r' recursive).

20) MV (source) (destination)   (cut paste & rename the content).
{ ex :- MV hi ansible. (cut paste) to rename goto the directory than MV hi hi.sh }

Deleting content :-

21) rm (file name) } To remove files.
22) rm *.txt

23) rmdir   (TO remove empty directories).

24) rm -r -f (directory name)   (TO remov directories with content).
(or)
rm -rf (directory name).

Find something on linux :-

25) find (Location) -type (file/dir) -name (file/dir name)
-name

i) find . -type f -name (file name). (if searching for file).

ii) find / -type f -name *.log.

iii) sudo find / -type f -name *.log (* entire system).

(/ is entire system).

(sudo - super user privilage).

* if access denied issue than sudo privilage.

26) find . -type f -empty (To clean empty files).

27) find . -type d (directory name) (To find directories).

username.	encry pass.	user id	group id	description	home location	shell.
ec2-user	x:	1000:	1000:	EC2 default user	/home/ec2-user	/bin/bash

28) grep (word) (file). (find a word on a file (or) Group of files).

29) grep -w (word) (file) (To find exact file).

30) grep -w -i (word) (file) (ignore case sensitivity)

31) grep ^(word) (file). (^ at starting position of record).

32) sed 's/word/replace name/' filename (stream editor) (find and replace word in file)

ex:- sed 's/shutdown/break down/' passwd.

i) sed 's/word/replace/' filename (only first word).

ii) sed 's/word/replace/g' file name (all words).

iii) sed 's/word/replace/2' file name (Particular place of word)

↳ (place of number)

iv) sed -n 1,5p filename (pull particular records from file).

(number above 1,5)

ex:- sed -n 1,5P passwd.

v) sed numberd filename (To delete particular records from file).

ex:- sed 1-5d passwd.

=> This changes are not present on the actual file.

ex:- sed 's/shut down/break down/g' Passwd > passwd1

so create file using '>' write on passwd1 copy and the past this using 'cp' command.

→ cp passwd1 passwd

vi) sed -i (for directly applying the changes).

ex:- `sed -i 's/games/james/g' passwd.`

→ `Sudo find / -type f -empty` to find empty files of System

33) crontab

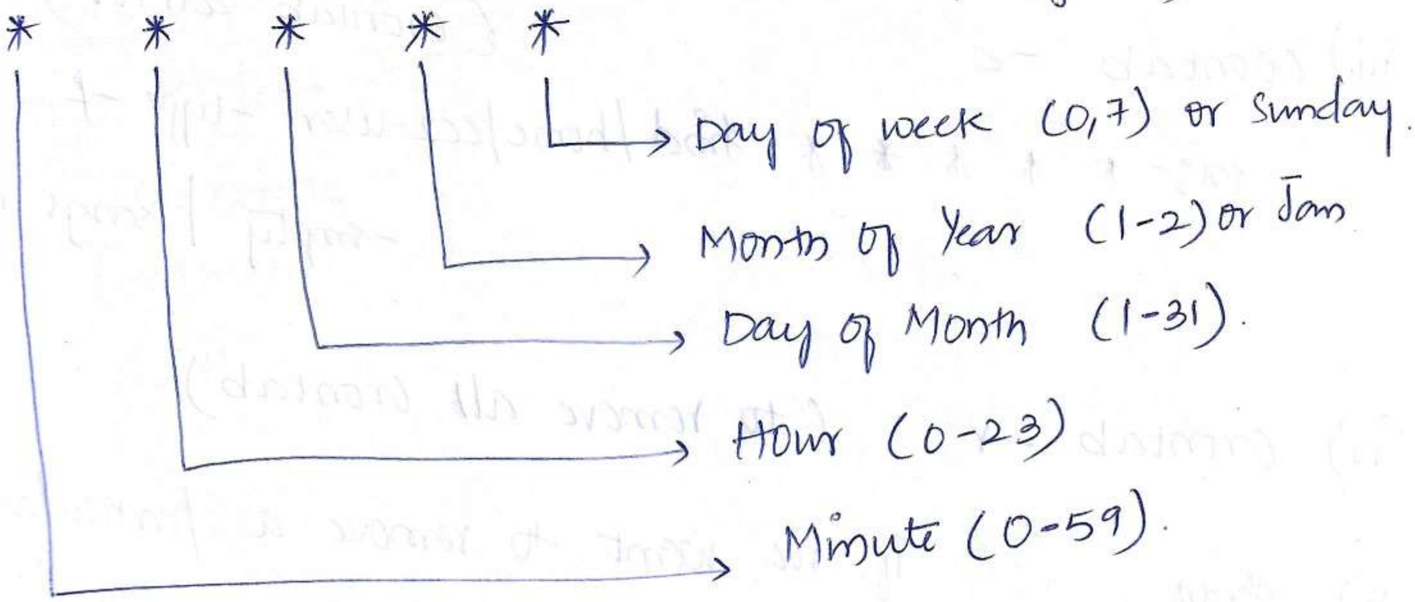
(Automate any task).
(this is schedual job)

ii) `crontab -L` (To see any Present crontab).

iii) ex:- (To clean up).

```
1) sudo find . -type f -empty
2) Sudo find . -type f -empty | xargs rm.
```

("|" for multiple commands Pipe symbol).



command = /Path/to/script.sh

Example :-	Min	Hour	DM	MY	DW.
Every min	*	*	*	*	*
Every day 9:30Am	30	9	*	*	*
M-f Every day 11.30pm	30	23	*	*	(M-f) or (1-5)
9AM to 6 PM	0	9-18	*	*	*
10AM fist Quarter	0	10	*	1-3	*
4PM Alternate moths.	0	16	*	1-12/2	*
1st Jan only	0	0	1	1	*

(or) @ monthly

iii) crontab -e (crontab editor).

```
ex:- * * * * * find /home/ec2-user -type f -empty | xargs rm
```

iv) crontab -r (to remove all crontab).

v) cron if we want to remove a particular crontab then open editor crontab -e and the delete using dd delete option.

** connecting :- 07/02/22

windows linux

Putty

laptop ←——————→ Remote server

Git bash

username /password

linux ←——————→ linux

SSH

* create another instance

1) use same key ex:- Vamshi-deveops

linux

Remote

⇒ connect Remote server same as linux server

connecting to one machine to another :-

Two ways:-

i, user name
Password

ii, SSH

i) Username &
Password → (103)
oldway :-
(server) ←

cd /etc/ (System related files are under /etc/ folder).

ls.

cd ssh.

Pwd

Now, we will find sshd_config.

sudo vi sshd-config (to update properties)

↓

Password Authention no.

→ insert mode — (i).

Add # at P.A no and (# to commento) in P.A.
remove # at P.A Yes

34) service sshd restart → (Failed).

sudo service sshd restart

35) To setup password for user

sudo passwd ec2-user

Password :- admin.

⇒ (101) (server) ⟵

ssh ec2-user@ IP.4 address.

Yes

Password : admin.

remember

<u>Steps</u> <u>to</u> <u>[illegible]</u> :-

i) On the remote.

/etc/ssh.

sshd-config.

Password Authentiōn Yes.

ii) service sshd restart.

iii) setup the Password for the
user whom we want to connect.

(ii) **S.S.H** **Mechanism:-** (New way)

(101) Server.

1) user.
2) Private key.
3) Public key.

36) sudo (useradd) (username) (TO creat user).

37) cat /etc/Passwd (To see who is the user).

sudo (useradd) (username).

38) su (username).
Password : (control c)

→ set up Password ←

39) sudo Passwd (user name). (Passwd is the Password command)

New pass :
Retype Pass :

su (username).
Pass :

→ Switched to user ←

Pwd
whoami

cd
pwd
whoami

→ user changed ⟵
⇒ Now create Private & Public key.

40) ssh-keygen. (To create ~~Password~~ key. for any user).

Enter.
Enter.
Enter.

ls -a.

→ .ssh created ⟸

cd.ss/.
ls.

→ Private and public key are created ⟵
cat
~~adt~~ id_rsa
cat id_rsa.pub.

Remote server :- (SSH-copy-id)
(135).

cat /etc/passwd.

~~So~~ Sudo (useradd) (username).

Sudo passwd (username).

N Pass :-

R Pass :-

Sudo vi /etc/ssh/sshd_config.

→ insert ← (i)

Add # to A.P NO.

Remove # to P.A Yes.

sudo service sshd restart.

* (101) Server *

su (username).

Password :

cd

ls -a

→ still now no .ssh ⇐

* (101) server *

ssh-copy-id (username)@ I.P address of (135).

Yes.

Password :

→ connected ←

* (135) Server *

ls -a

⟶ .ssh file created ⟵

cd.ss/.

ls.

⟶ authorized_keys ⟵ is created.

cat authorized_keys. (Public key).

* (101) server *

ssh (username)@I.P of 135.

⟶ connected ⟵

Manual way :- (204) Server

cat /etc/Passwd.

sudo useradd (username)

NP :-

R P :-

su (username).

Pwd

cd

Pwd

Mkdir .ssh.

ls -a

→ .ssh created.

cd .ss/.

ls

touch authorized-keys.

ls

→ authorized-keys created ←

vi authorized_keys

(copy ssh-rsa) from server (~~101~~ 135).

⇒ cat .ssh/id-rsa.pub ⇐

Paste the key in (204).

cat authorized_keys.

ls -l

→ Permissions ←

example :-

rw- - - - - - -

owner	group	others	u - user
rw-	- - -	- - -	g - group
			o - other
			a - all

- (Read) - 4
- (write) - 2
- (Execute). - 1

Check if the servers (135) and (204) have the Same Permissions.

41) chmod 600 authorized_keys.

{ chmod RWE (filename) }

ls -l

ssh (username) @ IP address.

Check the .ssh permissions with ls -la

Yes

→ Connected ←

42) ** SCP (content) username@I.P:/home/username
file

GIT

9/02/22

installations :- (Cent OS)

Check Git version.

43) git -- version.

44) Yum install git.

su (user name).
cd
Pwd
Whoami
Yum install git

→ not installed ←

sudo Yum install git

→ Not Git installed ← { Local user }.
(not installed since not a sudoers)

exit

ec2-user
Pwd
ec2-user.

sudo Yum install git { ec2-user).

→ Git installed ←

To provide admin privilages to users :-

i, user level

ii, Group level

cd / etc

ls

→ sudoers file is present ⟸

sudo vi sudoers.

cat group → ec2-user is a part of Group(wheel) so, is able to perform activities.

To, Give access to user :- (user level).

Sudo vi sudoers. (Not working)

Sudo -i

Pwd

vi /etc/Sudoers.

⇒ (copy using commands yy to copy and then paste 'p' root All=All All and paste and give (username))

⟶ not working ⟵

→ check Permission ←

ls -l /etc/sudoers. (file is having only
Read permission
i.e (r--r-----)

chmod 640 /etc/sudoers (i.e, rw-r-----)

ls -l /etc/sudoers

vi /etc/sudoers.

→ (copy "yy" root All=(All) All and paste "p"
username All=(All) All)

Hence the permissions are changed in the sudoers.

→ Now, re-turn to the first permissions).

chmod 440 /etc/sudoers.

ls -l /etc/sudoers.

su (username).

cd

pwd.

whoami

sudo Yum install git (Permission accessed).

→ Git already installed ←

Group Level :-

[useradd / usermod / user del] user level.

[Groupadd / group mod / group del] group level.

sudo groupadd deveops [can ~~create~~ add any group by using group add]

sudo usermod -aG wheel (username)

[git --version.

version 2.32.0.]

Yum install httpd

Service httpd

Service httpd status

Service httpd start.

Service httpd status.

cd /var/www/html/.

~~servicek~~
{ "Security group" click.

'Action'

E inbound ←

E outbound

Add rule.

HTTP

IPV4

[save rules]

* winscp Download :-

To copy data from linux to window (or) from windows to linux.

* Pick IP address
* winscp paste it

Advanced

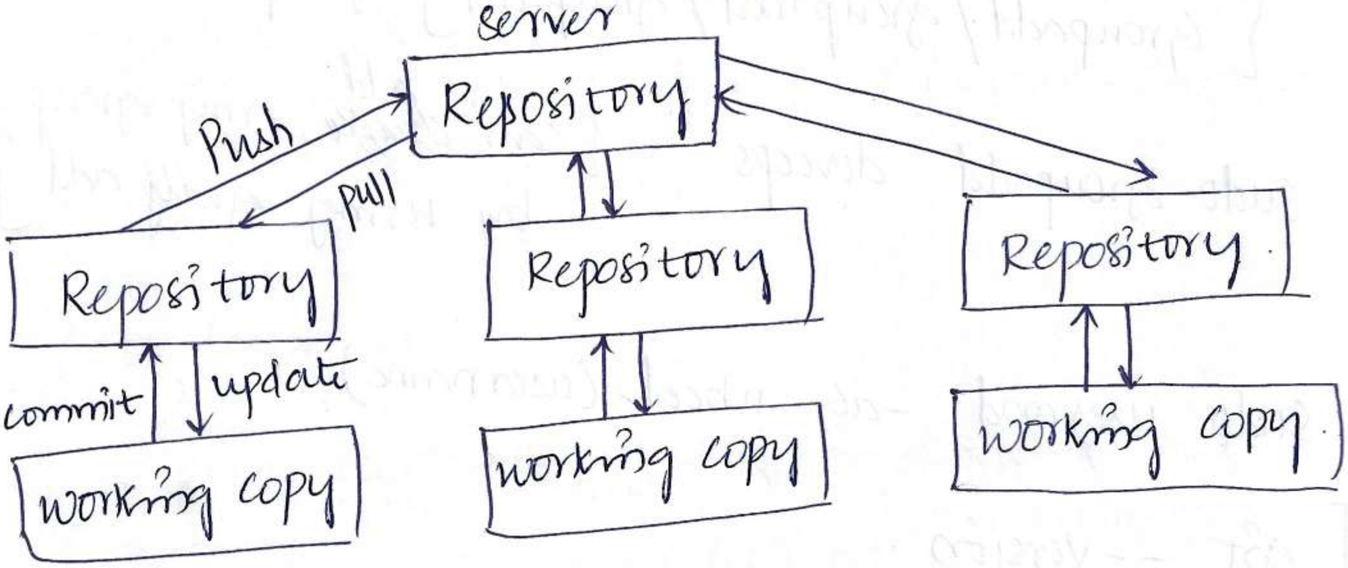
Advanced.

SSH AUTH Private key

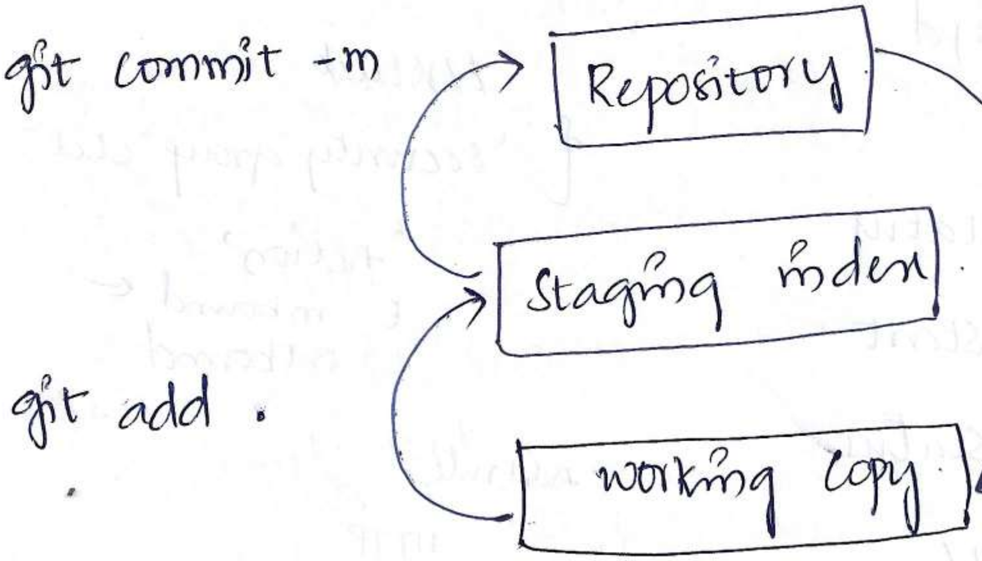
Version control tool:-

GIT :- it is Distributed Model

git is Distributed version control system.



GIT ARCHITECTURE :-



Code share-ability b/w the teammates & to manage History.

i) GIT (Local).

ii) GITHUB (Central)

iii) GITHUB ENTERPRISE (Rarely used). Paid one.

Sudo -i {root user}.

whoami

Yum install git

45) git --version

mkdir git-repo

ls

→ cd git-repo/

Pwd

git init → (to convert to repo) ←
to install

Pwd

ls ⇒ when git instalised it created a hidden directory called .git/

ls -a

cd .git/

ls

cd ..

46) git status

touch Sample.txt

ls -a

cd .git/

ls

cd ..

git status).   (to know the status).

47)git add .   (. current Location).

ls .git/

→ (index staging area)

cat .git/index.

git status

git branch.

→ (No branch because no commit is made).

48) git commit -m "created sample.txt"

49)git branch.

→ (Master branch is created).

ls.

git status.

ls.

vi sample.txt

→ Added some text ←

git add.

git status

git commit -m "three lines added"

0)git log → (To see commit id's and all
the commit History).

```
git config --global user.name "user name"
git config --global user.email "user email"
ls
vi sample.txt
  → Add text ←
git add.                         } git commit -am "message".
git commit -m "message           }
git status
git log.
```

GIT HUB :-

Working with GITHUB.

Github.com → Sign up.
→ required details.

Create a repository.

Owner / Repository

gitrub-repo. → centralized repository.

Created a repo.

→ Now take centralized repo data to Local repo.

Go to account.

open the repo. {ex:- gitrub-repo}.

⇒ HTTPS process (copy the path).

→ Local work station ←

51) sudo -i

ls

51) git clone (path). (copy the path from GitHub).

ls

cd git-repo/ (Local repo).

52) git remote -v (if we get any path then it is a central repo).

cd ..
cd gitub-repo/

* git remote -v (if we check again we can see a path, Hence it's a central repo).

⇒ instead reffering the whole path we can take origin. (Default).

touch login.java (placing some code)

ls
git status
git add.
git commit -m "login file created".
git branch

* main ——→ (Branch main is created
on GITHUB it is 'main'
on Local (Git) it is 'master')

git log ——→ (Histor is created).

git status

53) git push ——→ (To push changes from Local to Remote (Central).

user name:-
Pass word:- } (git GITHUB account details).

→ Access denied ← Pass-Ath is remove.

so, use token from GITHUB.

Go to Githmb.
settings.

* Developer setting.

* Personal access tokens. → Generate new token.

*Note

gitnb - token.

Expiration.

90 days.

* Generate the token.

*

- in HTTPS,
copy the token otherwise if we loose the token we have re-generate the tokens.

Go to local :-

user name : GITHUB username

Password : Token.

git log.

GO TO Github :- Check the github everthing is updated (Branch).

git pull → (To confirm both local and central are in sync).
or to syncranize).

vi login.java → (Add comment).

cat login.java

git add .

git commit -m "message"

git log.

cat login.java

git push

user name:

password : Token.

54) git pull = Fetch + merge.

SSH Mechanism :- (GITHUB)

in SSH :- NO username :-

and NO password :- (i.e:-TOKen)

SSH:- connecting to remote server securely. The process is on the control machine you need a user and for user you have to generate Public and private Key and on the Remote machine to the user you have to copy the public key.

or

on the remote side if you have the public key of user you can take the private key then straight forward with private key we can connect.

```
cd
ssh-key            (Create a Private & public key)
Enter
Enter
Enter
cd .ssh/
ls
cat id_rsa.pub     ( cospy this public key & paste in GitHub app )
```

→[To copy the public key from Github account.

Git Hub

settings.

* SSH and GPG Keys.

(create) - New SSH key.

ex:- root-key.] ←

```
mkdir ssh.
ls.
cd ssh/
ls.
git clone (paste)          (SSH (option) copy the
                            path).
ls
cd github-repo/
git remote -v.
ls.
touch readme.txt.
ls
git status
git add .
git commit -m "readme file added"
git push.
git log.
```

touch mytack.txt

ls

vi mytack.txt (Add (or) Edit comment).

git status.

55) touch .gitignore. (will ignore the content).

ls

ls -a

vi .gitignore

→ (add file for which content you want to ignore).

git status

git add .

git commit -m "message".

git Push.

56) git diff (compare the changes (b/w branches, commits, Local and remote).

57). Ps -ef (what are the services are running

Ps -ef | grep httpd (to know a particular service running (or) not.

Ps -ef | grep (service name)

58) netstat -nltp (network ports running).