

[GJIT]



git -



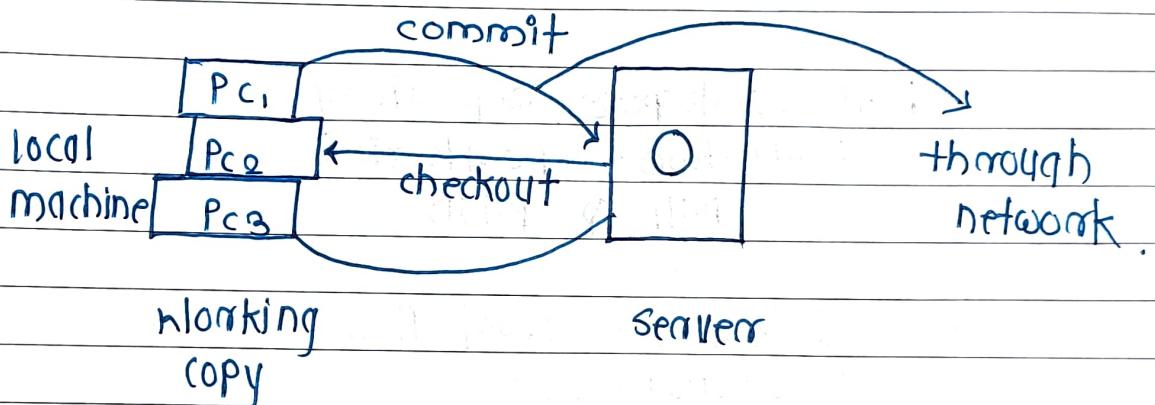
Version Control System

→

basically VCS is the software

To maintain the version, to tracking & sharing betⁿ team for that we need Version control system.

10



15

15



Distributed Version Control System -

↓

git is the example of it

→

git is tool

→

It is used for Source code management in software development

→

git is Version control system for tracking changes in computer files

local machine to

Server machine

use -

→

local machine to server machine pull, push

→

To merge the code

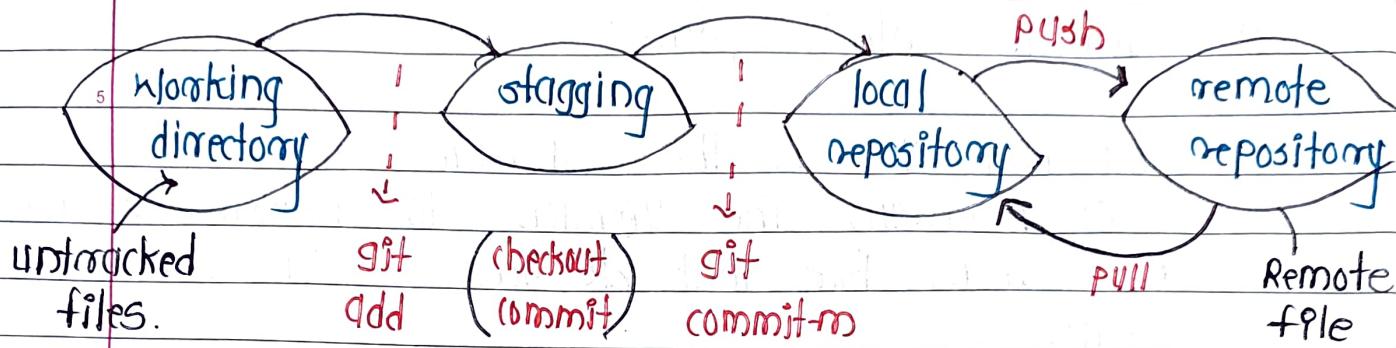
→

Code history maintain

→

Code security

Web application git - hub basically to communicate.



betw workspace - repository

betw two - repositories

PULL & PUSH

15 git architecture & work flow -
git commands

16 Version Control -

your daily task

create things

save things

edit things

Save the thing again. need version control

25 History tracking

create document

spelling correction

grammar

fix

color

chang

30 heritage
description

company logo

Task	Note	command
1) Tell git who you are	configure the author name & email address	git config --global user.name "kiran kale" git config --global user.email "kirankale998@gmail.com"
2) Create a new local repository		git init
3) checkout a repository	create working copy of local repository	git clone /path/to/repository
	for remote server, use:	git clone [username]@host: /path/to/repository
4) add file	add one or more file to staging	git add <filename> git add *
5) commit	commit changes to head. staging to local	git commit -n "msg"
	commit any files you have added with git	git commit -d
	commit any file you have changed	git commit -a
	Science	

task	note	command
6) push	Send changes to the master branch of your remote repository	git push origin master
7) status	List the files you have changed & those you still need to commit	status
8) 15 connected to a remote repository	If you haven't connected your local repository to a remote server, add the server to able to push to it	git remote add origin <server>
25	List all currently configured remote repositories.	git remote -v
9) 30 Branches	Creating new branch & Switch to it	git checkout -b {branchname}

task	Note	command.
→)	Switch from one branch to another	git checkout branch name.
5		
→)	List all the branches in your repo, & also tell you what branch your currently in.	git branch
10		
12	Delete the feature branch	git branch -d <branchname>
15	push the branch to your remote repository	git push origin
20	push all branches to your remote repository	git push --all origin
25	delete a branch on your remote repository	git push origin :<branch name>
10)	fetch from the remote repository fetch and merge changes on the remote server to your working directory	git pull.

task

note

view all the merge conflicts view the conflicts against the base file.

5

After you have manually resolved any conflicts you make the changesfile

10

tag

you can use tagging to mark significant changes such as a release.

15

committed if the leading characters of the largest change set

20

ID, up to 10 but must be unique. Get the ID using

25

30

commands

git diff

git diff -base <filename>

git diff <sourcebranch>

<targetbranch>

git add file name.

git tag 1.0.0 <commit ID>

git log

task

note

commands.

- 12) local changes.
- If you mess up, you can replace the changes in your working tree with the last content in head changes.
- already added to the changes instead as well as new file will be kept.
- Instead to drop all your local changes and commit, fetch the latest history from the server & point your local master branch at it, do this.

`git checkout-{filename}`

`git fetch origin`

`git reset --hard origin/master`

13)

search

Search the working directory for `foo()`.

`git grep "foo()`

- git is a fast and modern implementation of version control.
- git provides a history of content changes.
- git facilitates collaborative changes to file
- git is easy to use of any type of knowledge workers.

- ① git init myproject - initialization
- ② cd myproject - history (all)
- ③ git add. - to node
- ④ git commit -m - staging to local

e.g doug & dan working together on same project.

doug working →
git checkout master
git commit -a -m "my new logo"
git push (sending to central server)

dan working →
git checkout -b danfeature
git commit -a -m "my feature code"
git push origin danfeature.

doug →
git pull
git merge danfeature

* Advanced git

git add -p myReport

git commit -m "Added latest statistics"

git log --graph --decorate --all --pretty=oneline

* 10 get going with git -

git installer download



configure your

user name & email



create repository



git init project

(cd project)

create file.txt

git add file1.txt

git commit -m "my first commit"

25 there are two version control systems.

① centralized



connect with central system.

through network

offline we cannot collaborate.

e.g ① subversion

② team foundation server

Distributed



- have own copies
 - we can work when offline
- e.g git
mercurial

git is most popular



① free

② open source

③ super fast

④ scalable

⑤ It makes easy to share code.