PAGE NO. :
DATE : / /

Grit (Revision Notes)

Grit :- git is a code management tool

⇒ Local version control → version control is on the system of desktop
limitations of LVCS → chances of data loss
collaboration was not possible easily

⇒ Centralized version control system :- Centralized server is used for collaborative work

Limitations :- ① Internet connectivity is required for CVCS.
no internet no work possible

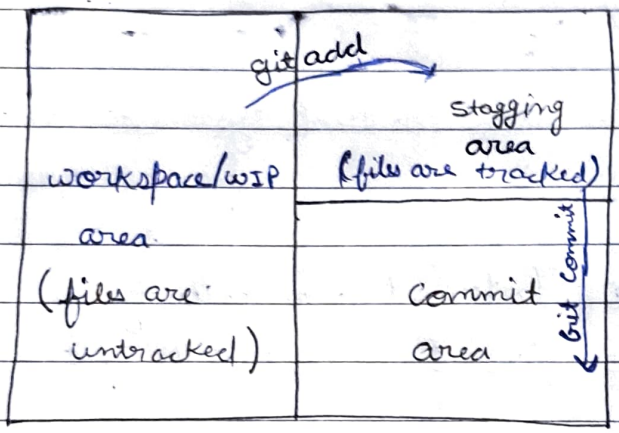
⇒ Distributed version control system (DVCS) :- ~~code is~~ version of code is present on both central server & on the local users laptop also.
eg :- Grit (most popular) (made by linux + ordinals)
Mercurial
Bit Keeper
Bazaar
Grit hub
GritLab
Bit Bucket
Aws code commit
Helix Core

we will work on Grit.

PAGE NO.:
DATE: / /

Benifits of Grit:- work on snapshots, hence less storage is used
Every operation is local → hence no need of internet
Grit has integrity → security is enhanced
Grit generally only add Data
Grit work on anysystem it is platform independent.

⇒ Yum install git → to install git on linux
#git init → to initialize git in a directory



git status → To check status of files or data which file is in which stage or file is tracked or not
if it shows nothing means we have commited all files.

git add filename → to start tracking of file or send in Stagging Area

git commit → to save / send in commit area

git commit -m "message"
option to add message

PAGE NO. :
DATE : / /

git log :- To show no. of commit with commit id & message
(unique ids for each commit)

git config user.name "Name" → To set your name
git config user.email "e-mail" → To set your email address.
This will only chang of the curret project or directory.

To change the name & email value for the current user (we will use --global option)

git --global config user.name "name"
git --global config user.email "e-mail"
} This will change for current user.

git clone http://github.com--- → to download a repo from github. we can specify the name by which it will show in our System
↳ it download any repo Untill it is a public repo.

git remote add origin http://github (Public repo link) → we have made a variable named "origin" here so that बार बार Link na likhna pde
↳ koi bhi naam rakh sakte
git remote → to check

git push -u origin master → To push krna is case me master ka data jayega
(using name we defined above)
↳ username :- github username
Password :- we have to use SSH key or token (on github)

⇒ To configure so that it doesn't ask username password every time

st1 git config --global credential.helper store (for all project) (to store in a file)
(we can use cache also to store temporary)
st2 ↳ git push -u origin master
↳ enter username password for once

PAGE NO.:
DATE: / /

to check the username password } → it is stored in:-
"/root/.gitconfig"

⇒ Reset :- used to rollback the code

Soft :- file is available in staging & working area. reset in commit

Mixed :- file is available in working area. no data in staggin in commit

Hard :- data is not available in working, stagging & commit area

→ ~1 me ye 1 step niche aayega 2 me 2 ←

git reset --soft HEAD~1
--mixed
--hard

★ By default it will do mixed reset

to reset ^till a specific commit ↗ choose option as per choice
git reset --hard <commit id>

⇒ To ignore some specific file from commit.
make a file ".gitignore"
add the names of file in ".gitignore" { *.log
now it will ignore the file mention ↑

⇒ Branch :-

git branch } To check no of branch & current branch

PAGE NO.:
DATE: / /

git checkout -b KGF2 } This will make a branch and go to that / move branch
(KGF2 → new branch name; -b ↳ option to change to new branch)

to check use # git Branch

git checkout master }
(master → name of branch to move)

git merge KGF2 } to merge the Branch
(KGF2 → name of Branch to be merged)

★ Jo Branch banayega wahi merge krega.

⇒ Rebase:- if any changes are done in master Branch after the branch is made then the slave Branch is to rebase itself.

git rebase master. } To rebase and run by slave brach

⇒ Squash:- when we don't require commit version of the slave branch when we merge.

git merge --squash Branch1 . } to merge without commit History
(Branch1 → Branch name)

⇒ Merge conflict:- master will approve the merge conflict by opening and editing it and after editing or making changes we can add & commit the file

⇒ How to push a only slave Branch to github.

git push origin Branch name

★ above command to be run from the specific branch to be push.

recommended ↑

fetch :- ye code ~ ko le ke aayega phir merge se merge hoga

git fetch

git merge origin/master

→ Ye sirf data ko fetch krega github se, phir manually merge hoga

no recommended →

Pull (Branch)

git pull origin master

★ isme wo bina puche fetch or merge ek sath kr dega

★ Jab **master Baranch** ko push krenge to -u dalenge

Reset

→ Ye galti ko mita dega.
→ Commit log se hi hat jayega

Revert

git revert <commit id>

isme data log me show hoga but head piche aa jayega
★ ye galti ko dikhayega. pr ~~dikhega nt~~ bdh jayega

fork :- to take repo from github to your system / account

Git hooks :- ~~use~~ used to automate some command which we use repeatedly. so that we don't have to write the codes again & again

vim post-commit → #!/bin/bash
git push -u HERO master
} we write script

this code written in file will run after every commit