

Git & Github

- Git : Version Control System → popular
 - free & Open Source
 - fast & scalable
- Track changes in code
- Track history of project
- Collaborate team
- GitHub : Website that allows developers to store and manage their code using Git.
 - folder → Repository → Repo
 - md : mark down
 - changes → commit (save)

GitHub Account -

1. Create a new Repo : apnacollege-demo
2. make our first commit
 - file ← add changes
 - commit → file(changed)

Setting up Git : git --version

Required -

- VS code
- or
 - └ windows (Git Bash)
 - └ mac (Terminal)

Install Git

- Configure Git : Tell the git account of github in which you will make changes.

\$ git config --global user.name "MyName"

\$ git config --global user.name "MyName"

--global : for whole git belong to the email

--local : for specific repo or project change using different email.

\$ git config --global user.email "abc@gmail.com"

\$ git config --list

credential.helper

user.name

user.email

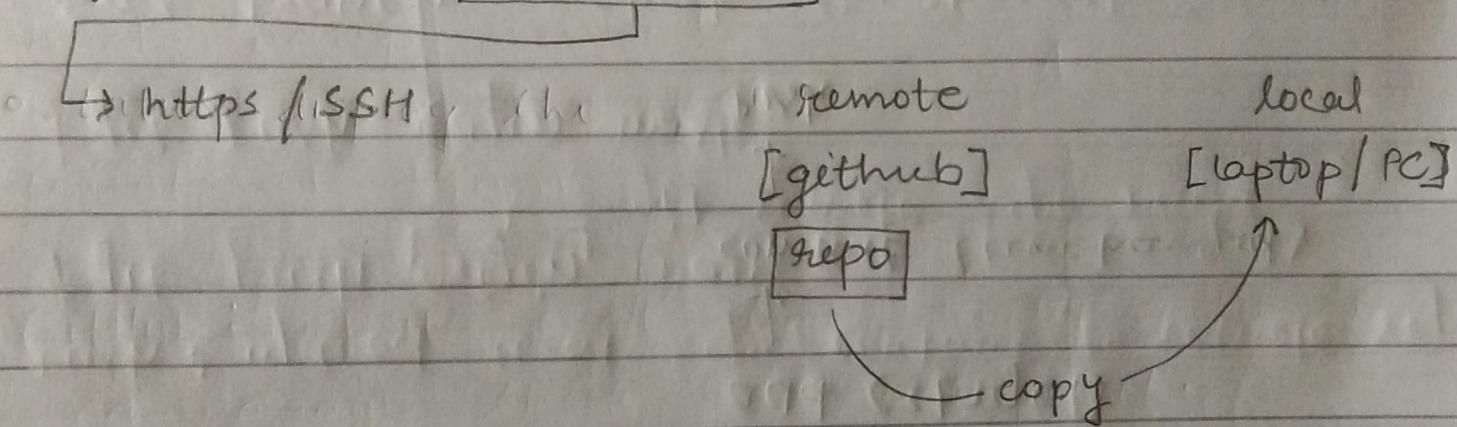
folder → Directory

Some Basic commands -

1. clone & status

o clone: cloning a repository on our local machine

\$ git clone <-some link->



- To get https or ssh link go to git repo > code > copy https/ssh link.
- Open cmd > \$ git clone <https/ssh link>
- Every repo have .git folder inside every directory.

o status: displays the state of the code

\$ git status

: up to date

: modified : <file name> [M]

: untracked file : <file name> [U]

19: modified file

Untracked file

- Types of status in git

1. untracked : new files that git does not track.
2. modified : changed
3. staged : file is ready to be committed
4. unmodified : unchanged

changed / new file
(modified) / (untracked)

↓
add (staged)

↓
commit (unchanged)

↓
Push

2. Add & Commit & Push

• add : adds new or changed files in your working directory to the Git staging area.

\$ git add <- file name →

\$ git add . : add all files present in cwd

• commit : it is the record of change

\$ git commit -m "some message"

• push : upload local repo content to remote repo

\$ git push origin main

 └── git remote repo name

\$ git push origin master

\$ git branch — check the branch name

◦ If project started from our local system then—

3. Init

init : used to create a new git repo

\$ git init

\$ git remote add origin <- link ->

\$ git remote -v (to verify remote)

\$ git branch -M main (to rename branch)

\$ git push origin main

 └ git remote repo name

◦ Steps—

1. Create a directory cpp-program-repo

2. use cd and go into directory cpp-program-repo

3. \$ git init

4. Create files as per your requirement.

5. \$ git add .

6. \$ git commit -m "Initial file"

7. go to github and create a repo & copy repo link

8. \$ git remote add origin <repo link>

9. \$ git remote -v

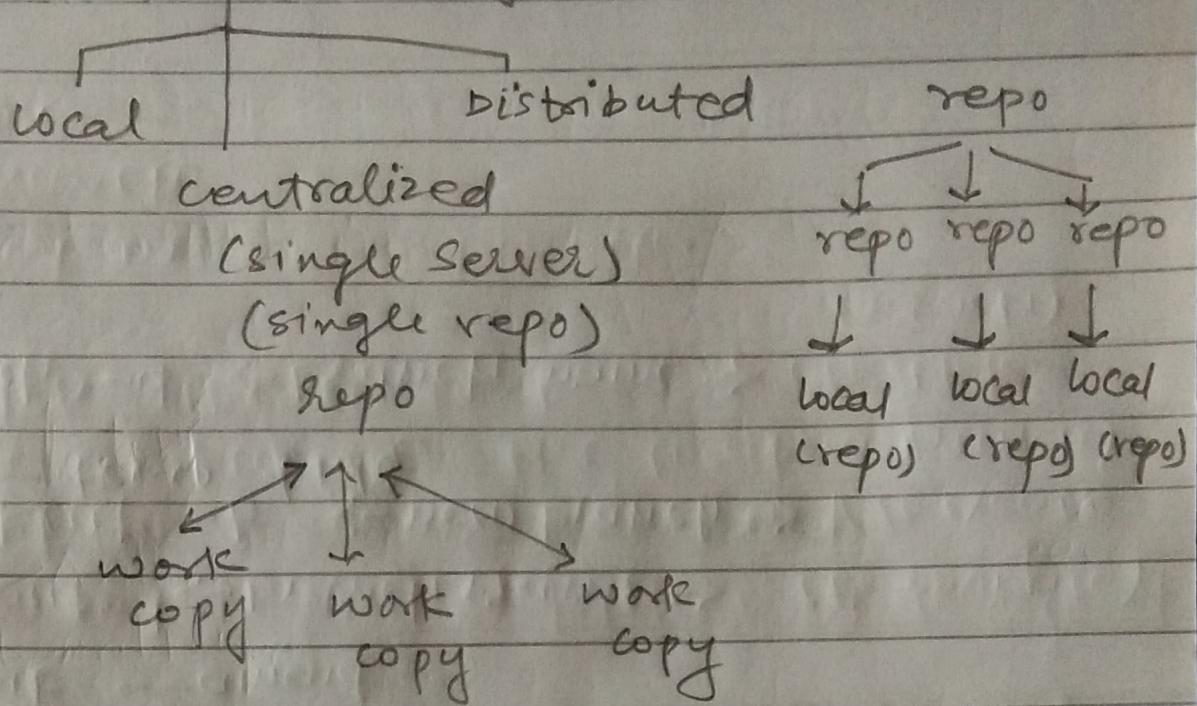
\$ git branch

\$ git branch -M main

\$ git push -u origin main

• Git

Types of Version Control System



• Create a repo

.git inside a repo

-- HEAD (current branch)

-- config (contain all configuration preferences)

-- description (description of project)

-- index (staging area b/w working dir & repo)

-- logs (keep records to changes and that)

- o .git directory

- .git
 - hooks (folder)
 - info (folder)
 - objects (folder)
 - refs (folder)
 - config (file)
 - description (file)
 - HEAD (file) : current branch

WorkFlow — git-workflow

- o Local Git

Github repo

clone

↓

changes

↑

add

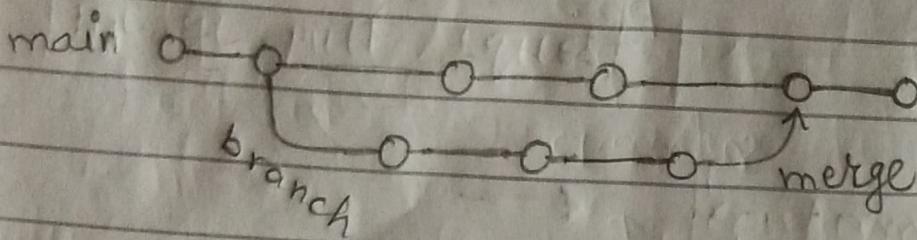
↑

commit

↓

push

Git Branches -



Branch Commands

\$ git branch (to check branch)

\$ git branch -M main (to rename branch)

\$ git checkout <-branch name -> (to navigate)

\$ git checkout -b <-new branch -> (to create new branch)

\$ git branch -d <-branch name -> (to delete branch)

• Git merging code

merge : To merge two branches

1. Way 1 -

\$ git diff <branch name> (to compare commits
branches, files & more)

\$ git merge <branch name> (to merge 2
branches)

1. check in which branch you are present

2. \$ git diff <other branch name>

\$ git diff main

3. \$ git merge main

2. Way 2 -

Create PR (Pull Request)

Pull Request : It lets you tell others about
changes you have pushed to a branch in a
repository on GitHub.

(main) → PR



SR Developer Review the PR

1. Go to Github select **Compare & pull request**

merge pull request

confirm merge

o Pull command —

`$ git pull origin main`

Pull command is used to fetch and download content from a remote repo and immediately update the local repo to match the content.

• Be present in main - branch

`$ git branch`

• merge conflicts Resolving

An event that takes place when Git is unable to automatically resolve differences in code b/w two commits.

\$ git checkout feature

\$ git merge main

\$ git checkout main

\$ git merge feature

\$ git push

Undoing Changes —

• Remember:

- If you merge 2 or more branch by pull request then use

\$ git pull origin main

OR

\$ git pull

into vs code to change make also in local repo.

- If you merge 2 or more branch by merge command

\$ git merge feature

then use

\$ git push origin main

OR

\$ git push origin main

into vs code and to change also make
on remote repo.

undoing changes

case 1 : staged changes

change add but not commit

\$ git reset <file name>

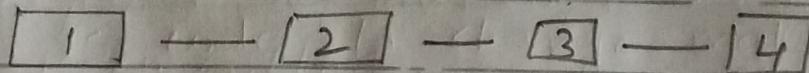
\$ git reset

- If mistakenly you added the change but not commit then to do undo use above command.

case 2 : committed changes (for one commit)

change add and commit

\$ git reset HEAD~1



HEAD~1 HEAD

- To check current commit status

\$ git log

Case 3: committed changes (for many commits)

\$ git reset <command hash>

\$ git reset --hard <commit hash>

hash: commit code

CURRENT COMMITTER (for one commit)

Fork

A fork is a new repository that shares code and visibility setting with the original "upstream" repository.

Fork is a rough copy