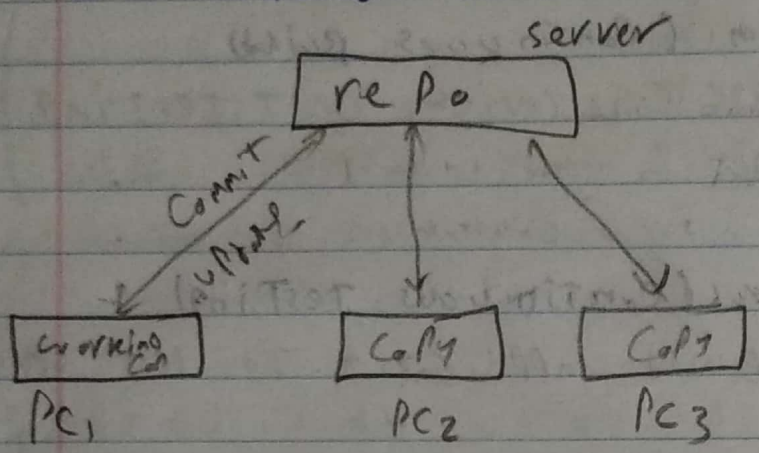
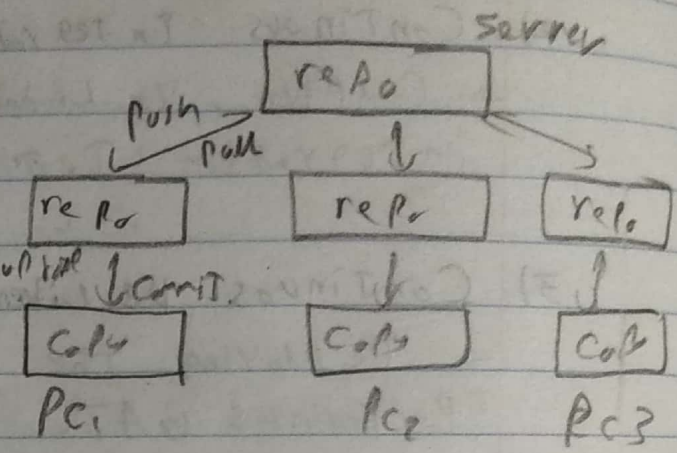
Source Code Management

⇒ Management of Changes To documents, Computer programs, large websites, other collection of info.

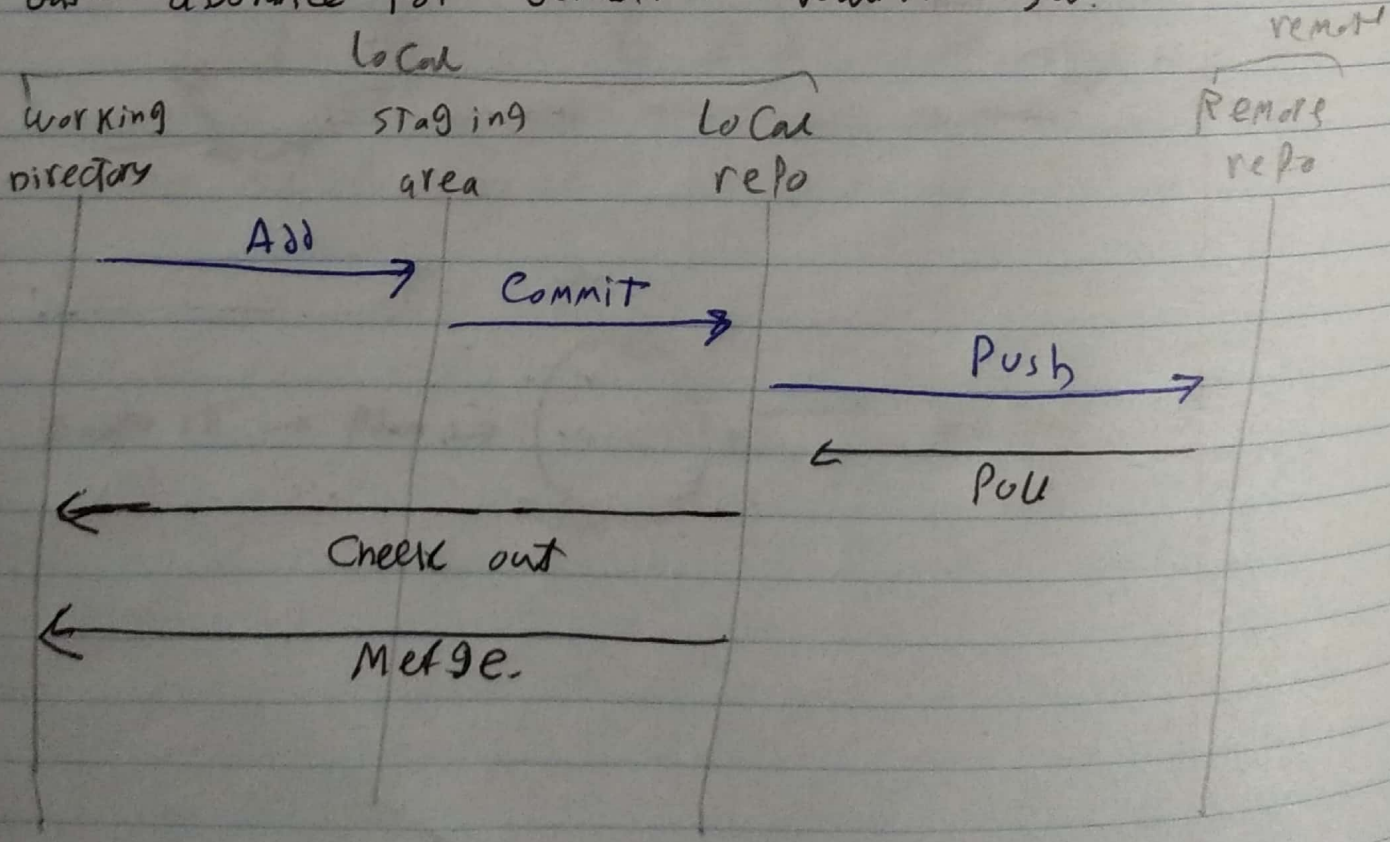
Centralized version



distributed version.



git ⇒ Git is distributed version control tool that support distributed non-linear work flow providing data assurance for developing quality sw.



GIT

(VCS) (SCM)

Version Control system / source Code manager.

⇒ it's an Tool That Manages different version of Co

Commit

Fundamental unit in GiT . snapshots of mini file system.

repo

working directory is the files That you see in your Computer's file system.

Checkout

when Content in repo Copied To working directory

Staging Area / staging index / index

A file in git directory That stores info about what will go into your next Commit.

prep table where git Take Next Commit

Secure Hash Algorithm (SHA)

basically ID Number For each Commit

Consit of 40 Char of 0-9 & a-f.

Branch

when new line of development Created That diverges From main line of development.

Creation

git init
Create new local repos on my Computer

git clone
Clone exist repos from some where
else to your local Computer

git status
Check status of repo

git log
Display info about existing Commits

git show (SHA) one
display info about given Commit (SHA)
all (Sov only

git log show → SHA
→ Author
→ Date
→ message -- one line

- git log --stat show log & edit files in that Commit

- git log -p , show log & the added & deleted lines of Code

- NOTE: You Can add them Together -- p --stat
with show
stat
then p

NOTE: git log SHA will show info about (SHA) Commit

git add file FILE — or . For all directory
add files from working directory To staging

git Commit
Take files From staging index & save them in repo

git diff
Display difference between Two version of a file.

- git rm -- Cached To remove files From staging area

* Note: git Commit will open editor To write your message other way
git Commit -m " message "

~~- git ig~~

- .gitignore : add Files names You don't want To add it when add all.

- git Tag
  Add Tag To specific Commits
  " extra Comment "
- git branch.
  Allow Multiple lines of development
- git Checkout branch name.
  switch between different branches & Tags
- git merge
  Combines Changes on different branches

* git Tag → -a Tag1 adding highlighted Tag
  → -d Tag1 remove the Tag

* You Can edit Tag To old Commit
  by providing its SHA
  git Tag -a/d Tag SHA

* git branch : its display current branch
  git branch name : create new branch.

* Note: You Can branch From old version
  git branch nof SHA of old Commit

* For deletion use -d ⇒ -D For force deletion
  git branch -d no b

* NOTE: You CaT delete branch You Currently on
  or have unique Commit.
  Use Force Deletion -D

name of new branch

- git checkout -b nab base branch
  create new branch & switch to it

- To display graph explain the branches

  git log --oneline --decorate --graph --all

- git commit --amend
  Alter the most-recent commit

- git revert SHA
  reverse given commit with it's (SHA) which means if line added will be deleted and etc~
  by creating new commit.

- git reset
  remove commit from repo * must be delete in same order

- when erase some thing by mistake using git reset you can get it again in 3.0 by using git reflog

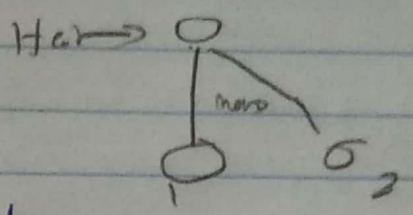
NOTES when want To refere To other Commit in Current Commit use Ancestry refrence.

^ ⇒ Parent Commit (which Parent in same level)

~ ⇒ First Parent

~2 & ^^ ⇒ Grand Pa (which Parent in Branch)

refere To Head pointer,



To Chose 1

HEAD ^ & HEAD ~1

To Chose 2

HEAD ^^ & HEAD ~2

- git reset modes

git reset mode Point To what

-- mixed (default)

move Commit To working directory.

-- Soft

move Commit To staging index

-- hard

move To Trash

- You Can Take Care and Take backup

git branch backup → create back up

at any problem git merge backup