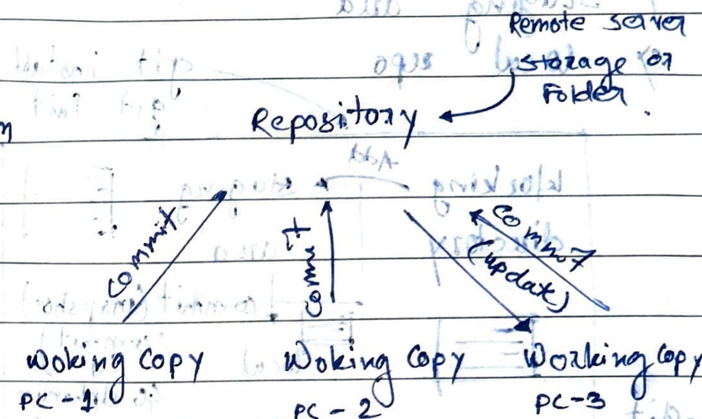
Page No.:
Date: 28/11/23

→ Guftgu lec-11

Introduction to Git - Linus Torvald (2005)

- Software Configuration Management OR Source code Management
  - Centralised version Control System
  - Distributed —"—

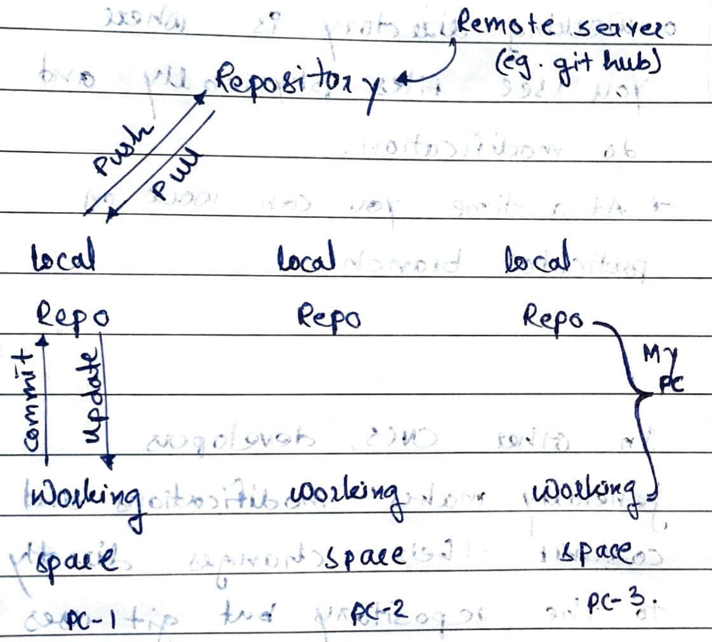
- Centralised version Control system (CVCS)



Drawback

1) It is not locally available. Need to be connected to network to perform any action (internet) always.
2) Since everything is centralised, if central server gets failed, you will loose entire data.
   eg: svn tool.

- Distributed version control System. (DVCS)



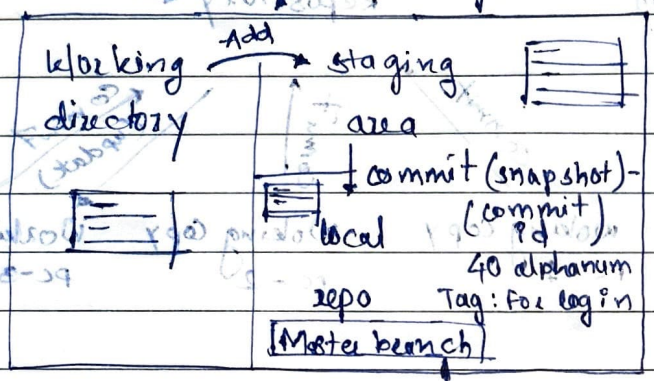
In DVCS, every contributor has local copy or 'clone' of the main Repository i.e. everyone maintains a repository of their own which contains all the files and metadata present in main repository.

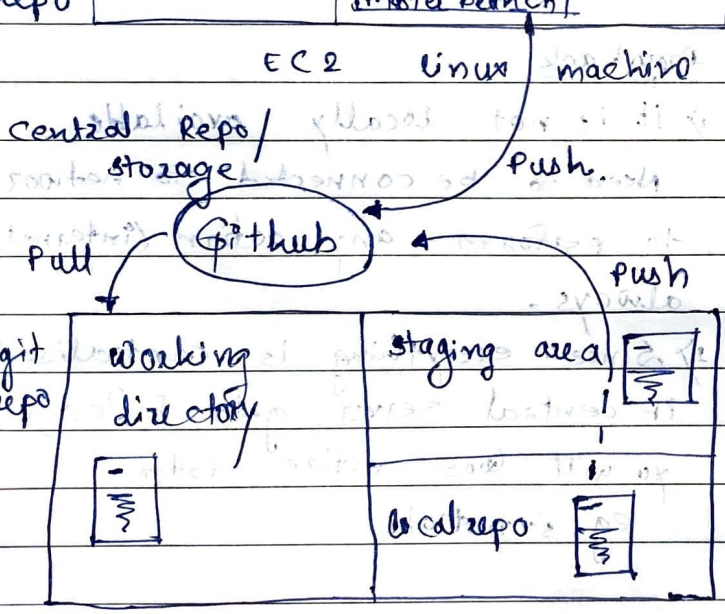
Git = software
Github = Service } Central Repo
Git lab = Service
Git ≠ Github, Git lab.

Gutjgu lec-12

Stages of Git and its terminology

1) Working directory / Workspace
2) Staging area
3) local repo





working directory
↓ Add
Staging Area
↓ Commit
Local repository
↓
Central Repository.

- Repository is a place where you have all your codes or kind of folder upon server
  - → It is kind of folder related to one product.
  - → changes are personal to that particular repository
- Server stores all repositories
  - → It contains metadata also
- Working directory is where you see files physically and do modification.
  - → At a time you can work on particular branch.

In other CVCS, developers generally makes modifications and commit their changes directly to the repository but git uses a different strategy. Git does not track each and every modified file. Whenever you do commit an operation git looks for the files present in the staging area only those files present in the staging area are considered for commit and not all the modified files.

- Commit
  - store changes in repository
  - you will get one commit id
  - 40 alpha-numeric character
  - Uses SHA-1 Checksum concept
  - Even if you change one dot, commit-ID will get change
  - Actually helps you to track the changes
  - Commit is also named as SHA1 hash.

SHA-1 Checksum helps in Integrity = Verification process

- Commit ID / Version ID / Version
  - Reference to identify each change
  - To identify who changed the file.

- Push

Push operations copies changes from a local repo. instances to a remote or central repo. This is used to store the changes permanently into the git repository.

- Pull

Pull operations copies the changes from a remote repo. to a local machine. Used for synchronisation between two repo.

- Tag
  - Tag assign a meaningful name with a specific version in the repository
  - Once a tag is created for a particular save, even if you create a new commit, it will not be updated.

- Snapshot
  - Represents some data of particular time
  - It is always increamental i.e. it stores the changes (appended data) only, not entire copy.

- Branch

Product is same, so one repository but different task
- Each task has one seperate branch
- Finally merges (code) all branches
- Useful when you want to work parallel
- Can create one branch on the basis of another branch.
- Changes are personal to that particular branch.
- Default branch is master
- file created in workspace will be visible in any of the branch workspace until you commit. Once you commit then that file belongs to that particular branch.

M T W T F S S
Page No.:
Date:
YOUVA

Types of repositories

Central repo = Bare repo:
→ store & share only
→ local repo = Non Bare repo.
→ Can modify the file

Advantages of git.

→ free and open source.
→ Fast and small.
→ Security - git uses a common cryptographic hash function called secure hash function (SHA 1) to name and identify objects within database

Gurftgu lec - 13

→ sudo su
→ yum update -y
→ yum install git -y
→ git --version
→ git config --global user.name "Aks"
→ git config --global user.email "..."
→ git config --list

Gurftgu lec - 14

→ login to EC2 instance.
→ Create one directory
→ git init
→ Touch myfile (put some data)
→ git status
→ git add .
→ git commit -m "..."
→ git status
→ git log
→ git show <commit id>
→ git remote add origin <central git url>
→ git push -u origin master.

→ git pull -u origin master.
Create one directory
→ Make it Git
→ sync with github

M T W T F S S

Page No.:

Date: 30/11/23

YOUVA

→ To ignore some files while committing

→ Create one hidden file .gitignore and enter file format which you want to ignore.

for eg: vi .gitignore
(content) enter *.css
*.java

→ git add .gitignore

→ git commit -m "latest update exclude css"

→ git status

→ Create some text, java & css files and add them by running "git add."

for eg: touch file1.txt, file2.txt, file3.java, file4.css

→ ls

→ git status

→ git add .

→ git status

→ git commit -m "my text files only"

git log -1
↳ latest commit

git log -2
↳ latest two commit

git log --oneline
↳ all commits in one line.

git log --grep "ignore"
↳ To search commit by key words

M T W T F S S

Page No.: YOUVA

Date: 30/11/23

Guftgu lecture - 15

Git branch

↳ To see on which branch one is assigned.

* Master
= Master.

git branch branch 1

↳ To create new branch

git checkout branch 1

↳ To exit from master and enter branch 1

Git Conflict

when same (name) file having different content in different branches

If you do merge, conflict occurs (resolve conflict the add and commit)

Conflict occurs when you have to merge branches.

After merge and conflict check in vi editor.

→ To stash an item (only applies to modified files, not new files)

git branch -d branch 1

↳ To delete branch 1 → Delete

git branch -D branch 1

Pulling mechanism. (Merge)

git merge branch

↳ to merge in current branch (mostly master)

→ You can't merge branches of different repositories

→ We use pulling mechanism to merge branches

Git Stashing

Suppose you are implementing a new feature for your product. your code is in progress and suddenly a customer escalation comes. Because of this, you have to keep aside your new feature work for few hours. You cannot commit your partial code and also cannot throw away your changes. So you need some temporary storage, when you can store your partial changes and later on commit it.

M T W T F S S

Page No.: YOUVA

Date:

git stash

↳ To stash an item

git stash list

↳ To see stashed items list

git stash apply stash@{...}

↳ To apply stashed items

Then you can add and commit

git stash clear

↳ To clear the stash items

git reset

git reset is a powerful command that is used to undo local changes to the state of a git repo.

To reset staging area

git reset <filename>

git reset .

To reset the changes from both staging area and working directory at a time

git reset --hard

M T W T F S S

Page No.: YOUVA

Date:

Guftgu lec-16

git revert

The revert command helps you undo an existing commit.

It does not delete any data in this process instead rather creates a new commit with the included files reverted to their previous state so your version control history moves forward while the state of your file moves backward.

git revert <commit-id>

- Github clone
  - Open github website
  - login and choose existing repository
  - Go to your linux machine.
  - git clone <url of github repo>

  It creates a local repo automatically in linux with same name as in github.

- How to remove untracked files.

  # git clean -n (dry run)

  # git clean -f (forcefully)

- Tags

  Tag operation allows giving meaningful names to a specific version in the repository.

  - To apply tag.

    # git tag -a <tag name> -m <message> <commit id>

  - To see the list of tags.

    # git tag

  - To see particular commit content by using tag.

    # git show <tag name>

  - To delete a tag.

    # git tag -d <tag name>