GIT & GITHUB

#Git :- (developed by Linus Torvalds) in 2005.

- Git is an open-source distributed version control system.

What is Version Control System ?

→ Version Control is a system that records changes to a file or set of files over time so that you can recall a specific versions later.

Functions of Version Control System :

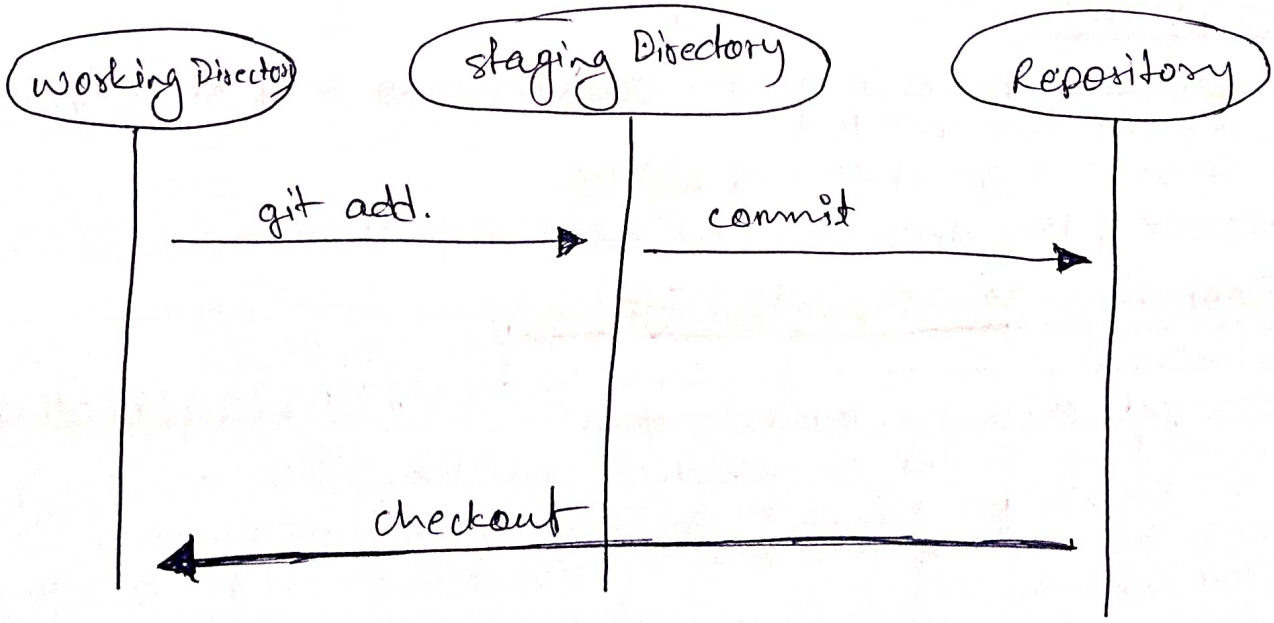
- Allows developers to work simultaneously
- Does not allow overwriting each other changes.
- Maintains a history of every version.

Some Features of Git

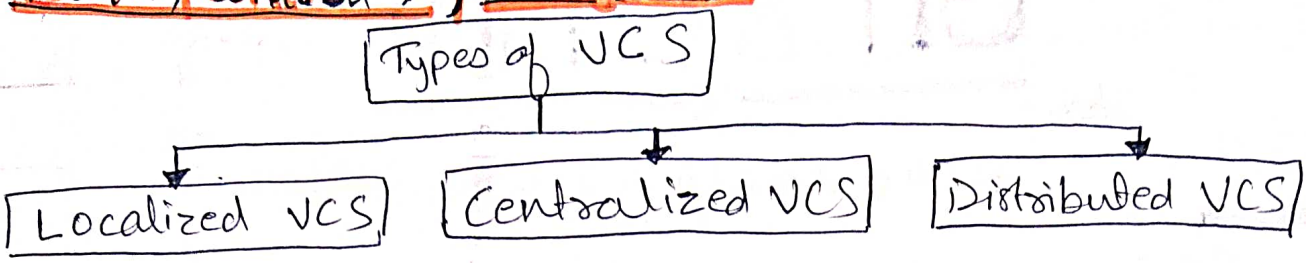
- Open-Source.
- Speed
- Easily recover files
- Rollback to previous working state.
- Track changes
- Saves Times
- Staging area

Benefits of Git :-

- Saves time
- Offline working
- Undo mistakes
- Track the changes.



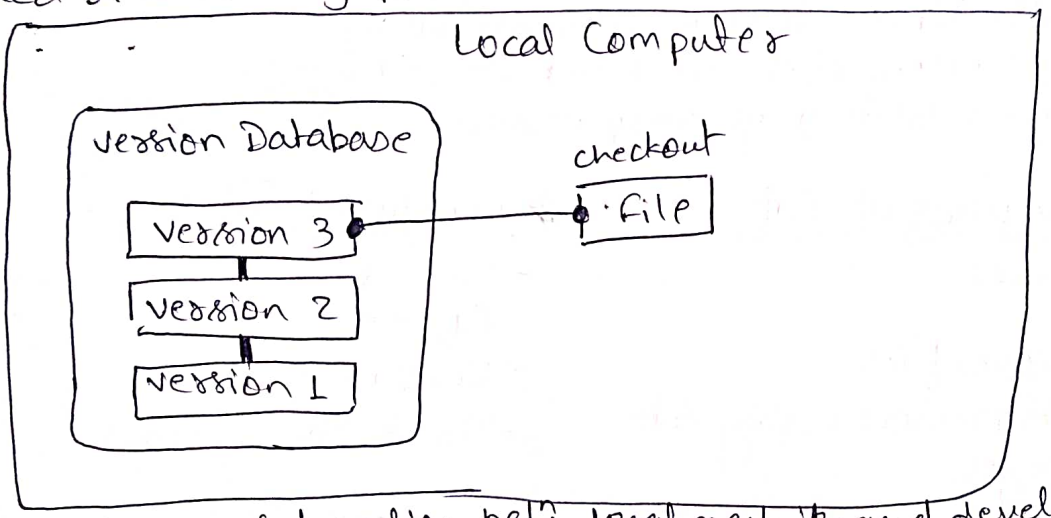
#Version Control System (VCS)



1 Local Version Control System :-

→ It is one of the simplest forms and has a database that kepts all the changes to files under revision control.

→ Revision Control System (RCS) is the most common VCS tools. It keeps patch sets. (difference between files) in a special format on disk. By adding up all the patches it can then re-create what any file looked like at any point in time.



* One to one interaction bet" local machine and developer which is a major drawback of this VCS.

Drawback :-

1. Stores all the data on the local computer so if at computer failure all the data will lost
2. There is high chance of errors.

→ Because of this drawback we have Centralized VCS

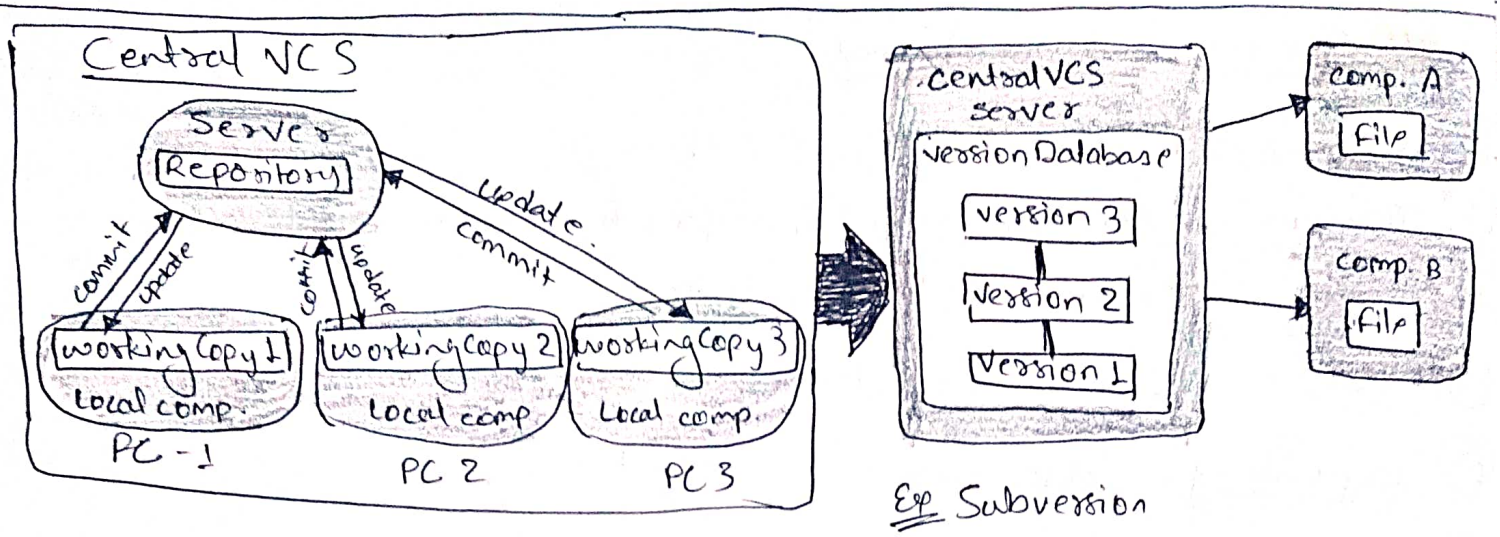
2 Centralized Version Control System :-

→ Centralized version control systems contains just one repository and each users gets their own working copy

→ You need to commit to reflecting your changes in the repository. It is possible for others to see your changes by updating.

→ Two things are required to make your changes visible to others are :
1. You commit
2. They Update.

→ The benefit of Centralized VCS is, make collaboration amongst developers along with providing an insight to a certain extent on what everyone else is doing on the project. It allows administrators to fine-grained control over who can do what.



Eg Subversion

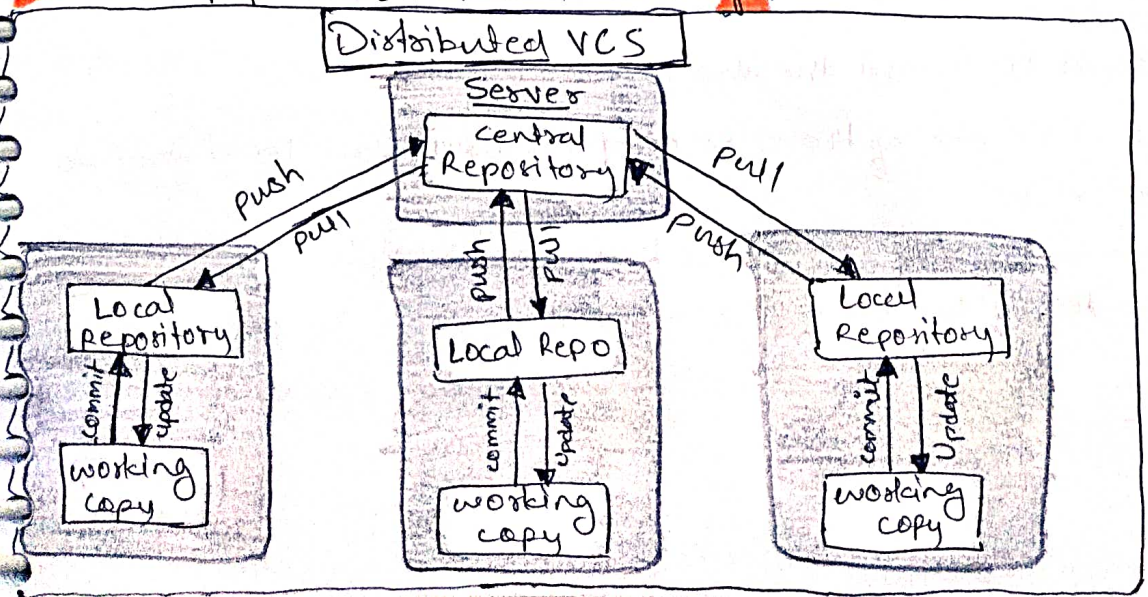
Drawback

- The single point of failure that the centralized repository represent if it goes down during that period collaboration & saving versioned changes is not possible.
  That means what happend if the server goes down or shut down then all the files is lost. and cannot Rollback

Therefore we have Distributed VCS.

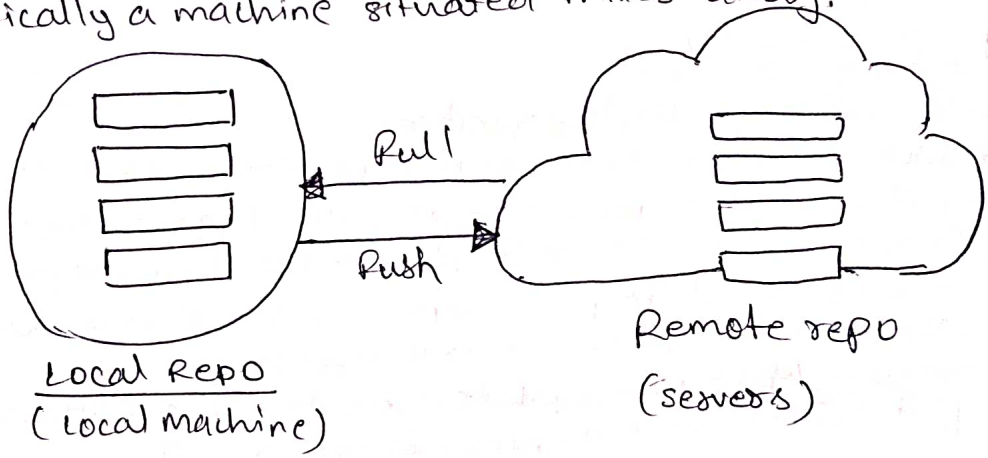
3 Distributed VCS :-

- It contain multiple repositories.
- Each users has their own repository and working copy.
- Just committing your changes will not give others access to your changes. This is because commit will reflect those changes in your local Repository and you need to push them in order to make ~~changes in your local~~ them visible on the central repository.
  Similary, when you update, you do not get other's changes unless you've first pulled those changes into your repository.
- In order to make changes visible to others, 4 things required :-
  1. You commit
  2. You Push
  3. They Pull
  4. They Update.
- most popular distributed VCS is git, mercurial.



Commonly Used Terms :-

1. Directory : Folder
2. Terminal or Command line : interface for text commands.
3. CLI :- Command line Interface.
4. Cd :- Change Directory
5. Repository :- Folder where the project is kept.
6. GitHub :- A website to host your repositories online.
7. Local Repository :- Git local repository is the one on which we will make local changes, typically this local repository is on your computer.
8. Remote Repository :- Git remote repository is the one of the servers, typically a machine situated miles away.



Git Commonly Used Terms :-

1. Clone : Brings a repo that is hosted somewhere like GitHub in Servers into a folder on your local machines.
2. Add : The git add command is used to add file contents
3. Commit : It is used to record the changes in the repo.
4. Push : Pushing is an act of transfer commits from your local repo to a remote repo.
5. Pull :- It fetches and merges changes from the remote server to your working directory. The git pull command is used to pull a repo.

And many more :- like fetch, merge, upstream, Branch etc.

GitHub :

→ GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git ~~plus its~~ plus its own features.

GitHub Important Terms :-

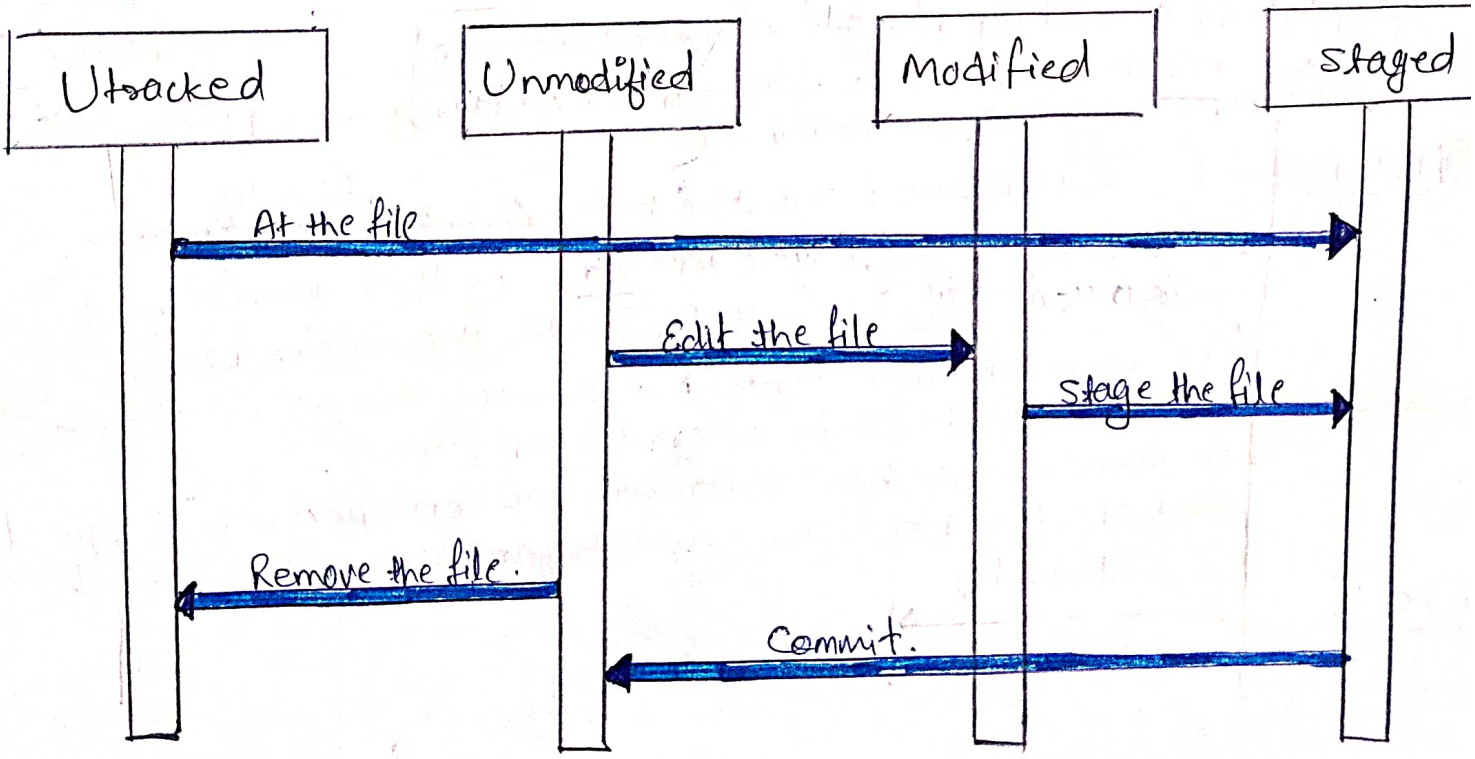
1. **Repository** : A GitHub repository can be used to store a development proje
   - It contain folders and any type of file (text, HTML, CSS, JS etc).
   - Basically it is a folder.
2. **Branch** :- A GitHub branch is used to work with different versions of a repository at the same time.
   - By default a repository has a master branch (a production branch)
   - Any other branch is a copy of master branch. (as it was at a point in time).
   - New branches are for fixing bugs and feature work separate from master branch, when changes are ready, they can be merged into the master branch. If you make changes to the master branch while working on a new branch, these updates can be pulled in.
3. **Commits** :- At GitHub, changes are called commits.
   - Each commit has a description explaining why a change was mad which is called commit msg.
4. **Pull Request** : Pull Request are the heart of GitHub collaboration.
   - With a Pull request you are proposing that your changes should be merged. (pulled in) with the master branch.
   - Pull requests show content differences, changes, addition, and subtractions in colors (green & red).
   - As soon as you have a commit, you can open a pull request and start a discussion; even before the code is finish.

Etc :-

Difference Bet^n Git & GitHub :-

Git	GitHub
① Git is a software.	① GitHub is service.
② Git is a command-line tool	② GitHub is a graphical user Interfa
③ Git is installed locally on the system	③ GitHub is hosted on the web.
④ Git is maintained by linux.	④ GitHub is maintained by microsoft.
⑤ Git is focused on version control and code sharing.	⑤ GitHub is focused on centralized source code hosting.
⑥ Git is a version control system to manage source code history	⑥ GitHub is a Hosting service for Git Repositories.
⑦ Git was first released in 2005.	⑦ GitHub was launched in 2008.
⑧ Git has no user management features	⑧ GitHub has built-in user management features.

Git - GitHub Architecture :-



Git 'diff' command :

→ Diff command is used in git to track the difference between the changes made on a file i.e. the difference between working directory and staging area.

Syntax

```
git diff <file name>
```

So we get

```
1. diff --git a/demo.txt b/demo.txt
2. index e69de29..a930f89 100644
3. --- a/demo.txt
4. +++ b/demo.txt
5. @@ -0,0 +1 @@
6. + Hello hi i am mahendra.
```

} we get as a output after running "git diff demo.txt"

Understanding meaning of this lines in Output (line-by-line)

1. diff --git a/demo.txt b/demo.txt
   - a/demo.txt → Source (The Staging Area)
   - b/demo.txt → Destination (The working directory)

2. index e69de29..a930f89 100644
   - e69de29. → Hash of file content from source/staging
   - .a930f89 → Hash of file content from destination/workspace
   - 100644 → Git file mode → 100 → Type of file (As it is txt file) (100)
     644 → file permision. (Read, write, Execute) (User, group, All)

3. --- a/demo.txt
   - source (The staging Area)
   - --- a/demo.txt → source file is missing some lines

4. +++ b/demo.txt
   - Destination (working directory)
   - +++ b/demo.txt → New lines are added in the destination file.

5. @@ -0,0 +1 @@
   - In the source file no lines are present
   - But in destination file one line is added

6. +Hello hi i am mahendra
   - " " → Space means Unchanged
   - "+" → Plus means lines are added to destination file (working directory)
   - "-" → minus means some lines are removed from destination file.