INTRODUCTION OF GIT

➔ Introduced in the year of 2005.

- **Author----** Linus Torvalds**.**
- **Written in ---** c, with programming scripts written in shell script and python**.**
- **Operating system ---** Linux, Mac Os, Windows**.**
- **License ---** GPL/ GNU [general public license].
- **Website ---** git-scm.com

WHAT IS GIT ?

- Git is open source tool.
- Distributed version control system.
- It is designed to handle minor to major projects with high speed and efficiency .
- It is developed to co-ordinate the work among the developers.
- Git is the foundation of many services like git hub and git lab**.**

WHY GIT ?

DATA INTEGRITY

- Git is developed to ensure the security and integrity of the content being versioned.
- While transferring data it always make sure that there is no data loss.

TRENDY VERSION CONTROL SYSTEM

- Git is the most widely used version control system.
- It has maximum project among all the vcs.
- Due to its amazing workflow and features.

EVERYTHING IS LOCAL

- All operations of the git can be performed locally.
- This is a significant reason for the use of git.
- We will not have to ensure internet connectivity.

COLLABORATE TO PUBLIC PROJECT

- There are many public projects available on the git hub.
- We can collaborate on those projects and show our creativity to the world.
- Many developers are collaborating on public projects.
- Collaboration allow us to stand with experience developers and learn a lot from them.

FEATUES OF GIT

- **Open Source**

- **Scalable** – When number of users get increased the git can easily handle such situations.

- **Speed** -- git is very fast, so it can complete all the task in a while, most of the git operations are done on local computer, so it provides high speed.

- **Maintains Clean history -- **it is one of the most helpful features of git. Git maintains a clean history of the project.

- **Distributed**

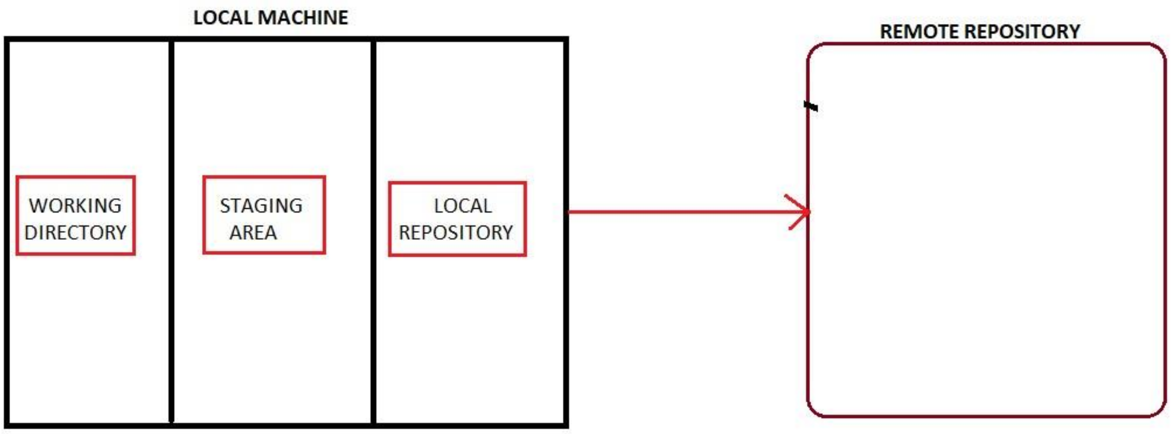
  →Means that instead of switching the project to another machine, we can create a copy of the entire project.
  →Also, instead of only having one central project, all the users having their own repository that contains all the files.
  →We do no need to connect to the remote repository.
  →Changes just stored in local repository, if necessary we can send these changes to the remote repository.

- **Customization is possible**

BENEFITS OF GIT

- Saves time.
- Offline working.
- Undo mistake.
- Track changes.

STAGES OF GIT / GIT ARCHITECTURE



Working Directory→ Here is the place where we do file management Operations such as – creating files, modifying files, deleting files.

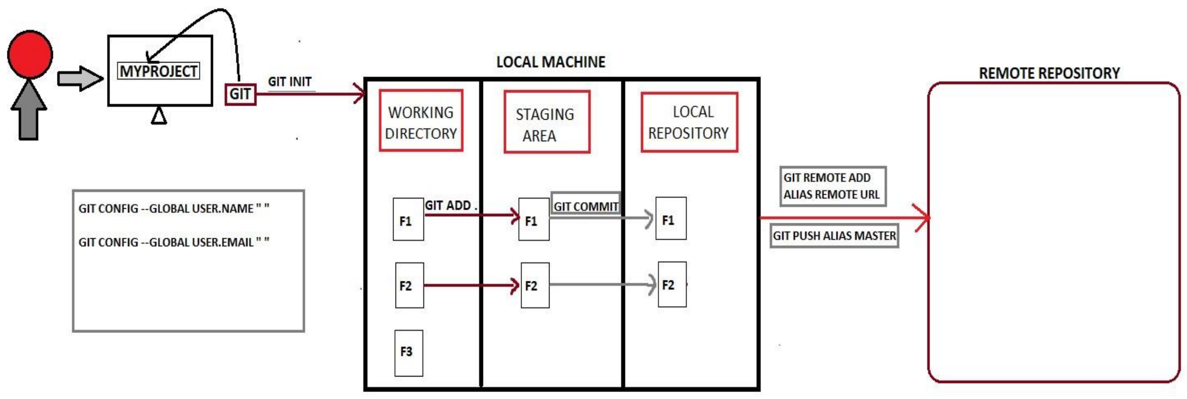
Staging area→ acts as temporary storage to save the files temporarily.

Repository→ nothing but folder → where we store our files and folders.

Local repository→ repository which present inside your laptop, pc.

Remote repository→ files and folders are stored in some remote location or some server is called remote repository.
Such as : git hub, bitbucket, git lab.

WORKING FLOW OF GIT



1. Create one folder.
2. Go inside of that folder.
3. Right click, open git bash
4. To initialize git inside this folder run **git init** command.
5. Do configuration by giving **user name** and **email.**
   run this command
   - git config - -global user.name “name”
   - git config - -global user. email “abc@gmail.com”

     or
   - git config user.name “name”
   - git config user. email “abc@gmail.com”

6. To move files from working directory to staging area
   run the following
   →To add specific file **git add filename**
   →To add all files **git add .**

→To add files with certain extension git add *.extension
Examples → git add *.java
→git add *.py

7. To move files from staging area to local repository **git commit -m “message”**

   To commit particular files
   git commit filename -m “message”

8. To see .git hidden files **ls -a**

9. To see the history of commits done **git log**

10. To link local repository to remote repository **git remote add alias [remote repository url**

11. To send files from local repository to remote repository **git push alias master**

DIFFERNCE BETWEEN GIT, MERCURIAL AND BAZAAR

GIT	MERCURIAL	BAZAAR
1. Git is an Open source tool	1. Mercurial is Open source tool	1. Bazaar is also an Open source tool.
2. Git provides more security	2. Mercurial will also provides more security	2. Bazaar provides low security.
3. Git is very fast.	3. Mercurial is little bit slow compare to Git	3. Bazaar is not much as fast as Git.
4. Git supports Branching and Merging.	4. Supports Branching and merging but not much as git.	4. Bazaar supports branching only upto some extent.
5. Git have Staging area	5. No Staging area	5. No staging area
6. Git maintains a Clean history of code.	6. Mercurial also provides clean history.	6. Bazaar will not provide clean history compere to git.