

GIT CLASS-1

GLOBAL INFORMATION TRACKER

- Git is used to track the files.
- It will maintain multiple versions of the same file.
- It is platform-independent.
- It is free and open-source.
- They can handle larger projects efficiently.
- It is 3rd generation of vcs.
- it is written on c programming
- it came on the year 2005

WORKING DIRECTORY:

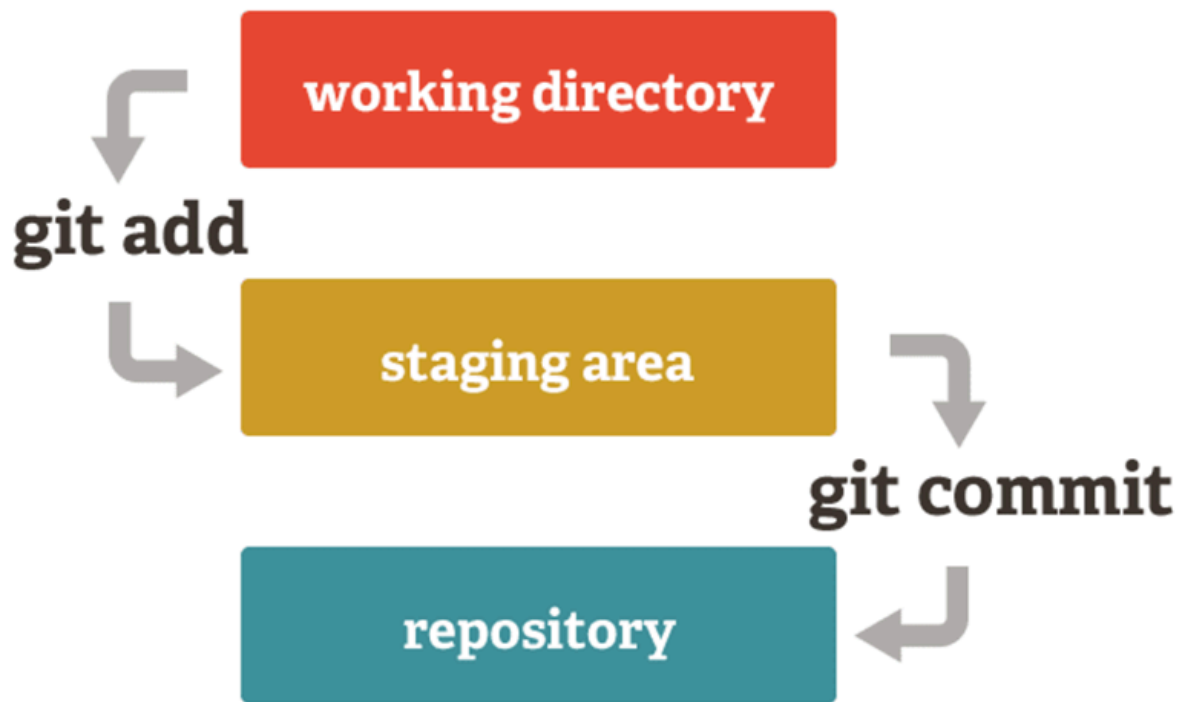
- In this stage git is only aware of having files in the project.
- It will not track these files until we commit those files.

STAGING AREA:

- The staging area is like a rough draft space, it's where you can git add the version of a file or multiple files that you want to save in your next commit.
- In other words, in the next version of your project.

REPOSITORY:

- Repository in Git is considered as your project folder.
- A repository has all the project-related data.
- It contains the collection of the files and also history of changes made to those files.



TYPES OF REPOSITORIES:

LOCAL REPO:

The Local Repository is everything in your .git directory. Mainly what you will see in your Local Repository are all of your checkpoints or commits. It is the area that saves everything (so don't delete it).

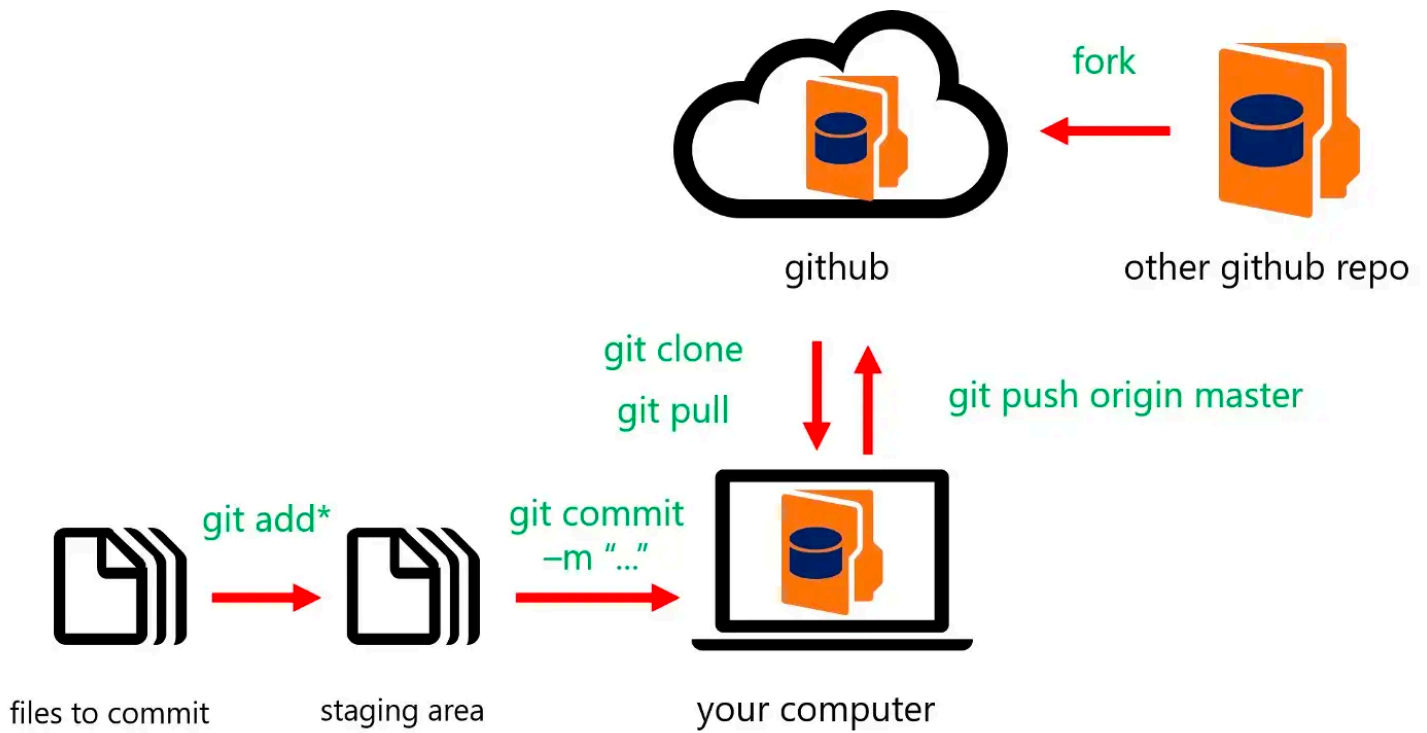
REMOTE REPO:

The remote repository is a Git repository that is stored on some remote computer.

GIT ALTERNATIVES:

- GIT LAB
- SVN
- BIT BUCKET
- P4
- STASH
- HELIX

GIT WORK FLOW:



TO INSTALL GIT : `yum install git -y`

yum = Yellowdog Updater Modifier

It is used to update, install or uninstall the packages

To check the version:

1. **`git --version`**
2. **`git -v`**
3. **`rpm -qa git`**

here rpm : redhat package management

-qa : query/verify all packages

To Initialise a git : `git init .`

Here init means initialise

. represents present working directory

when we initialise the git, .git folder will gets created which contains some files which relates to git.

TRACK & COMMIT A FILE:

- Create a file : **touch aws**
- Track a file : **git add aws**
- To check the status : **git status**
- To commit the file : **git commit -m "commit message" filename**

TRACK & COMMIT MULTIPLE FILES

- To track multiple files : **git add file1 file2 file3**
- To commit multiple files : **git commit -m "commit message" file1 file2 file3**

TRACK & COMMIT ALL FILES

- To track all files : **git add ***
- To track all files including hidden files: **git add .**
- To commit all files : **git commit -m "commit message" .**
- Here dot (.) represents all the files which are present in staging area

GIT LOG COMMANDS:

- To see the history of files : **git log**
- To see latest 2 commits : **git log -2**
- To see all commits-ids and messages : **git log --oneline**
- To see all full commits-ids and messages : **git log --pretty=oneline**

GIT SHOW COMMAND:

- To get the commit details along with filenames : **git show commit-id --name-only**
- To get the commit details along with the changes in filenames : **git show commit-id**
- To get the commit details along with the changes in files & stats : **git show commit-id --stat**

