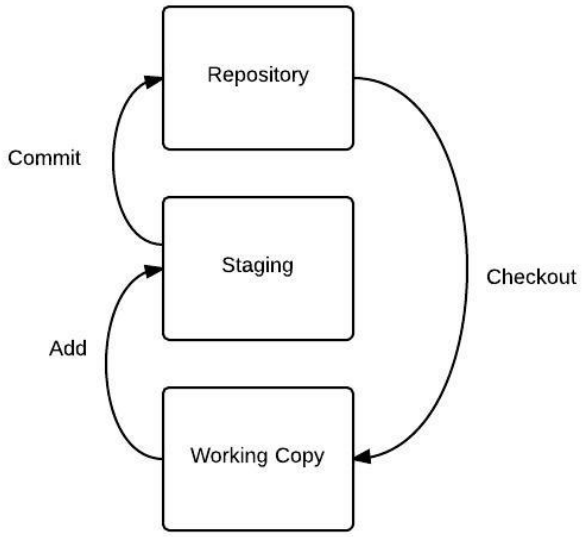
Why do we use Git:
Definition: Git is a version control system, a tool to manage source code history. Version control software allows developers to have "multiple versions" of a project, showing the changes that were made to the code over time, and it allows modify as needed.

Why do we need Git:

- **Git** makes it easy for different developers to work on a project at same time, using Git; multiple developers can work on files and easily merge their changes with the master branch of the project.

- An important thing about Git is that all the Git service is cloud-based file storage which is made possible through a remote repository. This remote repository stored on a server and a local repository is also stored on the computer of the developers working on the project. This feature of Git gives it an advantage in terms of speed

Explain the architecture of Git:



Typically a version control system has a 2 Tree system which is mainly repository and working copy. Now let us break it down to understand what each of these does.
Working copy: changes made on a project files are made here. Whenever you make an edit, it is saved in working copy and it is a physically stored in a disk.
Staging: This allows you to continue making changes to the working directory, and when you decide you want to interact with version control, it allows you to record changes in small commits.
Repository: all the version of the files or commits, logs etc is stored here in the repository. It is also saved in a disk and has its own set of files.
Checking-out: In order to make any edit on a file, it needs to be on your working copy and the file has to be gotten from the repository. This process is called Checking out. When you are done editing the file, you will save it back to the repository by *committing* it.
Committing: this is simply putting back the files from working copy to repository.