GIT

Git is a free and open source **distributed version control system** which is available for everyone at zero cost. It is developed to manage small to large projects with high speed and efficiency. It was created by ***Linus Torvalds*** in **2005** for the development of Linux Kernel.

Git is mainly developed to coordinate the work among the developers and it also allows us to track the changes made in files which is useful when we work together with our team members at the same workspace.

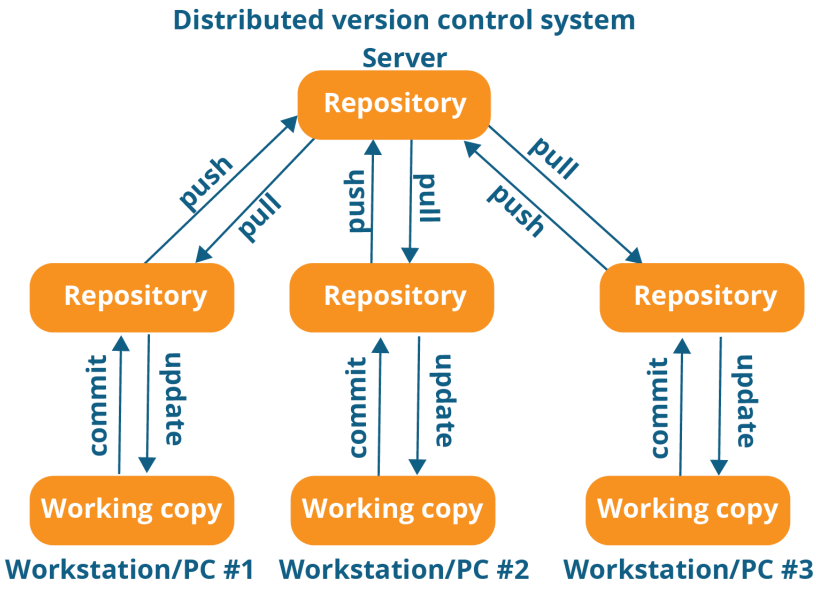
Why it is called as Distributed Version Control System ?

Version Control is the management of changes to documents, computer programs, large websites and other collection of information.

In a project multiple developers work on a same code and add code in parallel . So **Version control** helps in by maintaining a history of what changes have happened.

Git **repository** is used to contain all of the project files and entire revision history.

In Distributed Version Control System, every programmer maintains a local repository of their own which contains all the files and meta data present in the main repository(**central server**) . If you want to make any changes to the file in main repository you can use **pull** operation to download that file in your local repository and upload the edited file to the main repository by using **push** operation.



One of the main **advantage of GIT** is if the central server gets crashed at any point of time, the lost data can be easily recovered from any one of the contributor's local repositories.

Basic Git Terminology & Commands

GitHub is used for main or remote repository. It is an online service to which developers who use Git can connect and upload or download resources .

- **Git Clone**

  Git clone is used to clone an existing remote repository into your computer. If I want a local copy of my repository from GitHub , this tool allows creating a local copy of that repository on your local directory from the repository URL.

  `$ git clone URL`

- **Git Init**

  This command is used to create a local repository.

  `$ git init Demo`

- **Git add**

  Git add is used to add one or more files to the staging area.

  `$ git add Filename`

- **Git Commit**

  Committing is the process in which the code is added to the local repository. Before committing the code it should be in staging process. **Staging** is nothing but it keep tracks of all files which are to be committed. If the file is not in staging area then that file will not be committed.

  `$ git commit -m "Added print statement"` - Shows what code changes have done in that particular commit.

- **Git Status**

  The status command is used to display the state of the working directory and the staging area. But it do not show the information about committed project history.

  `$ git status`

- **Git Branch**

  A branch is a version of the repository that diverges from the main working project. A project can have more than one branch.

  Default branch in git is **Master Branch**.It is instantiated when first commit is made on project. Master Branch is the branch in which all the changes eventually get merged back. So that's why it is called as official working version of a project.

  `$ git branch` - This lists all the branches available in repository.

- **Git Merge**

  It is used to merge the new feature branch into the master branch.

  `$ git merge BranchName`

- **Git pull**

Pull is used to receive data from GitHub. Pull requests tells all the team members to review the code and merge that code(feature branch) into master branch.

```
$ git pull URL
```

- **Git Push**

Push is used to transfer commits from your local repository to a remote repository.

`$ git push [variable name] master` - sends the changes made on the master branch to remote repository.