**GitHub**

Github is a global repository system which is used for version control.

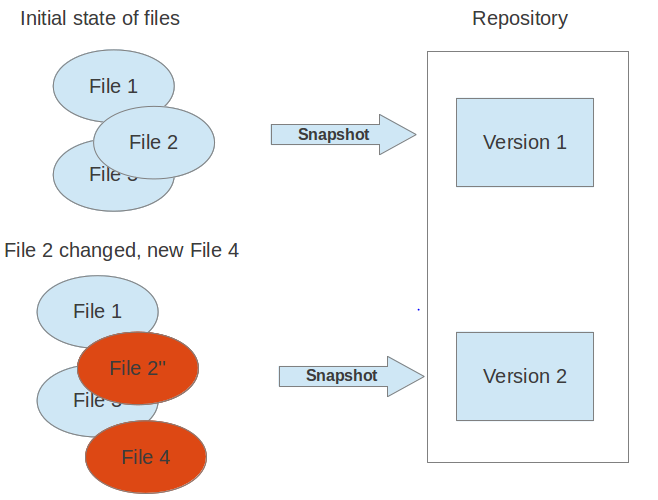
**Version Control System :**  is a software that helps software developers to work together and maintain a complete history of their work.

Listed below are the functions of a VCS:

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

Following are the types of VCS:

* Centralized version control system (CVCS).
* Distributed/Decentralized version control system (DVCS).



**Centralized version control system (CVCS) :**

Centralized version control system (CVCS) uses a central server to store all files and enables team collaboration. But the major drawback of CVCS is its single point of failure, i.e., failure of the central server.

**Distributed/Decentralized version control system (DVCS) :**

DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If the server goes down, then the repository from any client can be copied back to the server to restore it. Every checkout is a full backup of the repository. Git does not rely on the central server and that is why you can perform many operations when you are offline. You can commit changes, create branches, view logs, and perform other operations when you are offline. You require network connection only to publish your changes and take the latest changes.

### **Local Repository:**

### Every VCS tool provides a private workplace as a working copy. Developers make changes in their private workplace and after commit, these changes become a part of the repository. Git takes it one step further by providing them a private copy of the whole repository. Users can perform many operations with this repository such as add file, remove file, rename file, move file, commit changes, and many more.

### **Working Directory and Staging Area or Index**

The working directory is the place where files are checked out. In other CVCS, developers generally make modifications and commit their changes directly to the repository. But Git uses a different strategy. Git doesn’t track each and every modified file. Whenever you do commit an operation, Git looks for the files present in the staging area. Only those files present in the staging area are considered for commit and not all the modified files.

Let us see the basic workflow of Git.

**Step 1** : You modify a file from the working directory.

**Step 2** : You add these files to the staging area.

**Step 3** : You perform commit operation that moves the files from the staging area. After push operation, it stores the changes permanently to the Git repository.

