**Git**

Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by *Linus Torvalds* for Linux kernel development. Git is free software distributed under the terms of the GNU General Public License version 2.

**Version Control System(VCS)**

Version Control System is a software that helps software developers to work together and maintains the complete history of their work.

**Goals of VCS:**

1. Allow developers to work simultaneously.
2. Do not overwrite each other’s changes.
3. Maintain history of every version of everything.

**Types of VCS:**

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

## Distributed Version Control System (DVCS)

Centralized version control system uses central server to store all files and enables team collaboration. But the major drawback of CVCS is single point of failure, i.e., failure of central server. Unfortunately, if central server goes down for an hour, then during that hour no one can collaborate at all. And even in worst case if disk of central server gets corrupted and proper backup haven’t taken, then you will lose entire history of the project. Here, DVCS comes into picture.

DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If sever goes down, then repository from any client can be copied back to server to restore it. Every checkout is full backup of the repository. Git does not rely on central server 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 latest changes.

## Advantages of Git

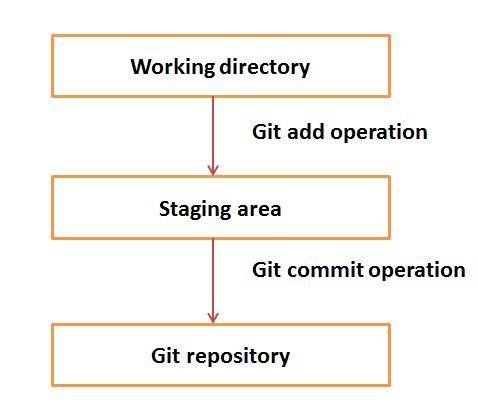
1. **Free and open source**
2. **Fast and small**
3. **Implicit backup**
4. **Security**
5. **No need of powerful hardware**
6. **Easier branching**

**Workflow of Git**

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

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

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



**Git Life Cycle**

1. You clone Git repository as a working copy.
2. You modify working copy by adding / editing files.
3. If necessary, you also update working copy by taking other developers' changes.
4. You review changes before commit.
5. You commit changes. If everything was fine, then you push changes to the repository.
6. After committing if you realize something was wrong, then you correct the last commit and push changes to the repository.

