
git

❖ VERSION CONTROL:

- **Version Control Software (VCS)** is also referred as **Source Code Management (SCM) tool** or **Revision Control System (RCS).**
- Version control is a way to keep a track of the changes in the code so that if something goes wrong, we can make comparisons in different code versions and revert to any previous version that we want.
- It is very much required where multiple developers are continuously working on /changing the source code.

WHY VERSION CONTROL?

- To Track different versions of a file or directory.
- Git tracks all text-based files like .html, .java, .jsp, php...etc.
- Not tracking Images, Videos, Audio files.... etc.

FUNCTIONS OF A VCS:

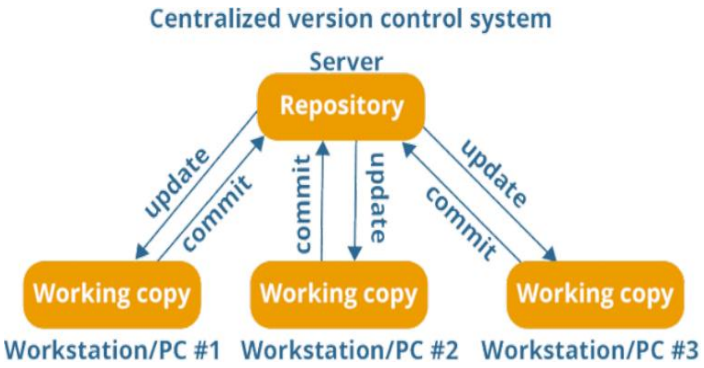
- Allows developers to work simultaneously.
- Does not allow overwriting each other's changes.
- Maintains a history of every version.

➢ TYPES OF VCS:

- Centralized Version Control System (CVCS)
- Distributed Version Control System (DVCS)

CENTRALIZED VERSION CONTROL SYSTEM (CVCS):

- Centralized VCS - CVCS uses a central server to store all files and enables team collaboration.
- CVCS works on a single repository to which users can directly access a central server.

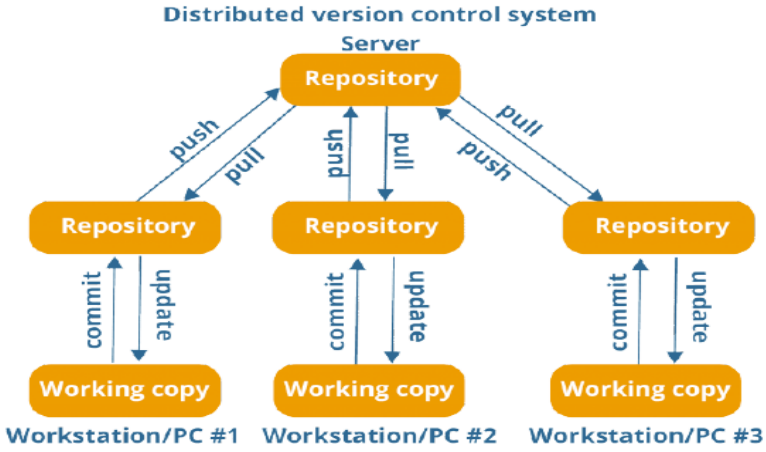


DRAWBACKS:

- It is not locally available.
- Crash of CVCS will result in losing the entire data of the project.

DISTRIBUTED VERSION CONTROL SYSTEM:

- In Distributed VCS, every contributor has a local copy or "clone" of the main repository.
- User can change and commit local Repo without any interference.
- User can update their local Repo from the Central Server.
- User can update the Central Server from their Repo.



- Operations in DVCS are fast.
- New changes can be done locally without manipulating the central data.
- 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.

➢ VERSION CONTROL TOOLS:

- Version control tools, also known as version control systems (VCS) or source code management (SCM) systems, are software applications that help track and manage changes to code and files. They are essential for development teams across all industries.
- The popular version control system tools are:
  - Git
  - SVN
  - Mercurial
  - Monotone
  - TFS
  - Visual SourceSafe
  - Revision Control System
  - CVS

❖ GIT:

- Git is a "**Version Control System" (VCS)** (or) **"Source Code Management" (SCM)** Tool.
- Git is a **Distributed/Decentralized** version control system, meaning your local copy of code is a complete version control repository. Most operations are local.
- Simple way to keep multiple versions of a file or directory.
- Repository contains files, history, config managed by Git.

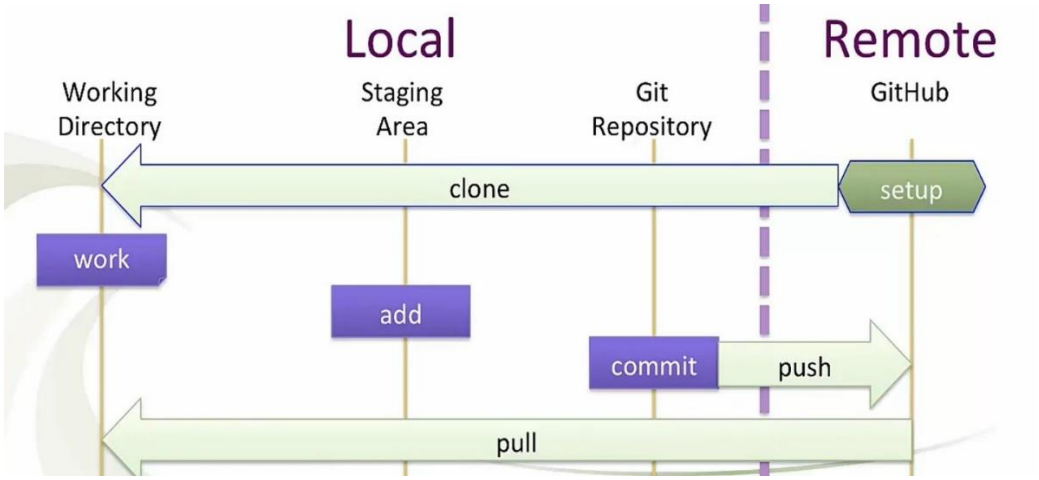
BENEFITS OF GIT:

- Free & Open source
- Simultaneous development
- Faster releases
- Built-in integration
- Strong community support
- Git works with your team
- Pull requests
- Branch policies

➢ WHY COMMAND LINE?

- There are a lot of different ways to use Git.
- There are the original command-line tools, and there are many graphical user interfaces of varying capabilities. For one, the command line is the only place you can run all Git commands — most of the GUIs implement only a partial subset of Git functionality for simplicity.
- History
- New Features
- Online Help
- Power!
- Consistent
  - Terminal on Mac/Linux
  - Git Bash on Windows

➢ GIT WORKFLOW:



GIT 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.
- Users can perform many operations with this repository such as add file, remove file, rename file, move file, commit changes, and many more.
- A GIT Repository contains Files, History, Config.

WORKING DIRECTORY: Area of Live files, also known as Untracked area of GIT.

STAGING AREA: Staging area is when git starts tracking and saving changes that occur in files.

GIT DIRECTORY: Also called 'Local Repo' is your **.git** repo. It's area where GIT save everything.

COMMIT:

- A commit is a snapshot of all your files at a point in time.
- One or more file changes.
- If a file has not changed from one commit to the next, Git uses the previously stored file.
- Commits create links to other commits, forming a graph of your development history.

BRANCHES:

- Each developer saves changes their own local code repository. As a result, you can have many different changes based off the same commit. Git provides tools for isolating changes and later merging them back together.
- Branches are lightweight pointers to work in progress, manage this separation.

REMOTE REPOSITORY (GITHUB):

- Remote Repository is stored on a code hosting service like GitHub or on an internal server.
- GitHub is a free, online platform for software development that allows users to store, track, and collaborate on projects.
- GitHub runs on Git, a version control system that helps programmers track changes in computer files. Users can also use GitHub Desktop, a free application that allows them to perform Git commands using a graphical user interface.