WHAT IS 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:

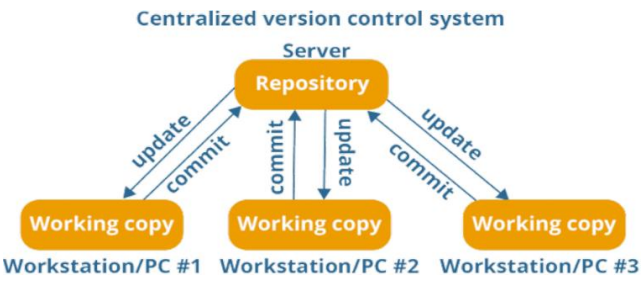
- Allows developers to work simultaneously.
- Does not allow overwriting each other's changes.
- Maintains a history of every version.

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.

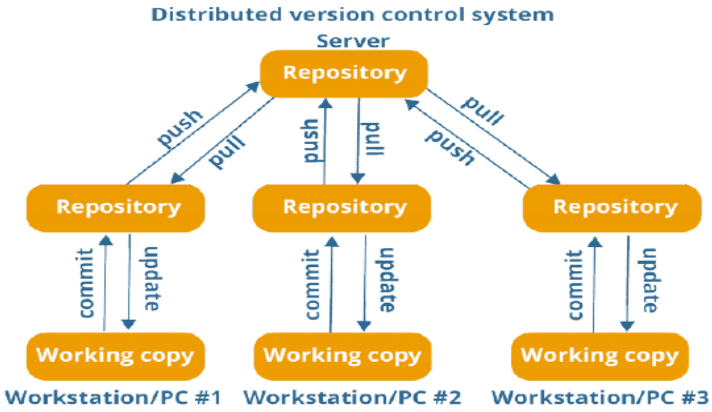


DRAWBACKS:

- 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:

- **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?

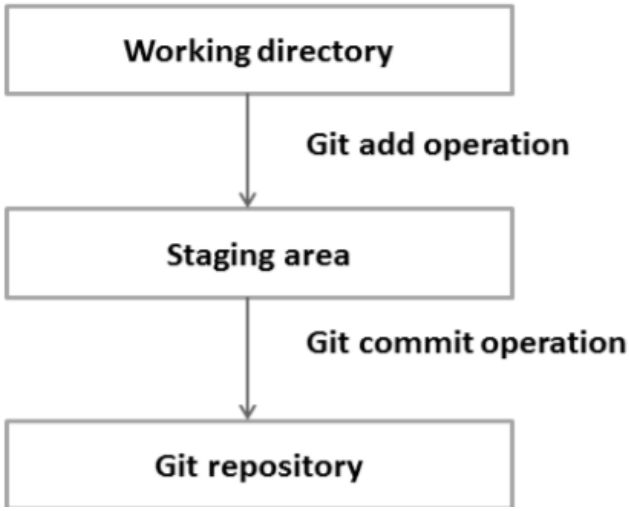
- History
- New Features
- Online Help
- Power!
- Consistent
  - Terminal on Mac/Linux
  - Git Bash on Windows

GIT TERMINOLOGY:

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.

STAGES OF GIT:



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.

REMOTE REPOSITORY (GitHub):

Remote Repository is stored on a code hosting service like GitHub or on an internal server.

GIT BASIC TERMS:

THE REPOSITORY:

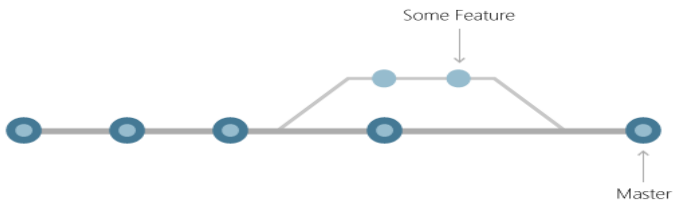
- Collation of files managed by git.
- Git can be initialized on a project to create a Git repository.
- A Git repository is the **.git/** folder that contains all your necessary repository files.
- It is a Working directory/Workspace.

THE COMMIT:

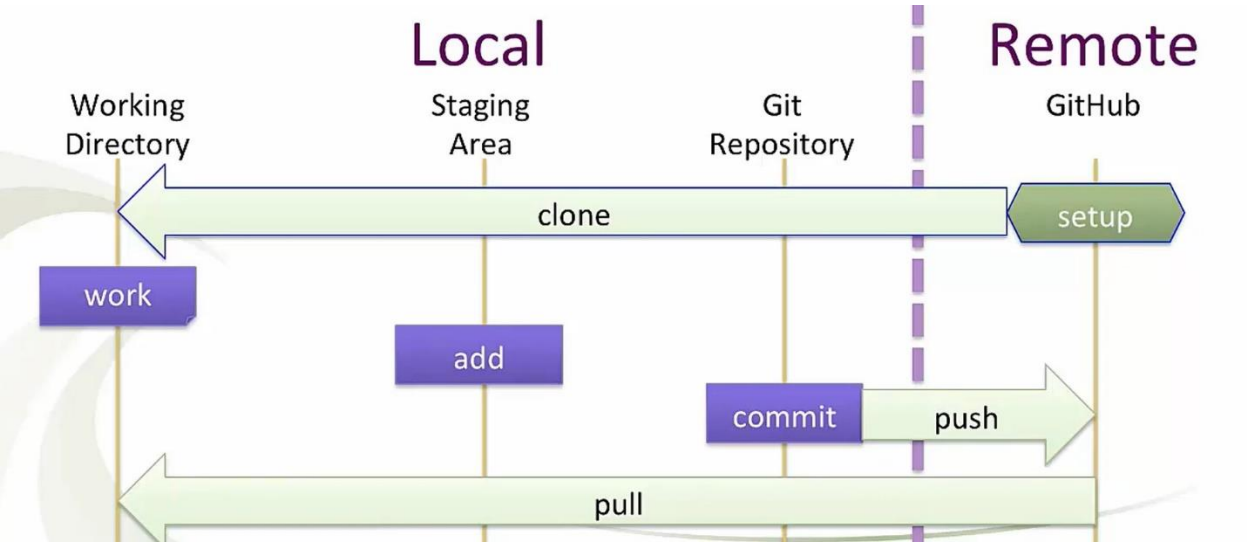
- A commit is a snapshot of all your files at a point in time.
- One or more file changes.
- If a file has not changed from one commit to the next, Git uses the previously stored file.
- Commits create links to other commits, forming a graph of your development history.
- Commits are identified in Git by a unique cryptographic hash of the contents of the commit.
- Commits on time line (Branch)

THE 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.
- Once your work created in a branch is finished, merge it back into your team's main (or master) branch.



GIT WORKFLOW:



GIT INSTALLATION:

- Before you start using Git, you have to make it available on your computer. Even if it's already installed, it's probably a good idea to update to the latest version.
- Git is an open source and can be installed on all major OS.
  - Unix / Linux
  - Mac
  - Windows

INSTALLING GIT ON WINDOWS:

- Download Git for Windows
- Browse to the official Git website: https://git-scm.com/downloads
  Click the download link for Windows and allow the download to complete.

$git config --global user.name "Ram"

$git config --global user.email "ram.ashokit@gmail.com"

INSTALLING GIT ON LINUX:

- If you want to install the basic Git tools on Linux via a binary installer, you can generally do so through the package management tool that comes with your distribution. If you're on Fedora (or any closely-related RPM-based distribution, such as RHEL or CentOS), you can use **dnf** or **yum:**

$sudo dnf install git-all -y

If you're on a Debian-based distribution, such as Ubuntu, try apt:

$sudo apt update -y

$sudo apt upgrade -y

$ sudo apt install git-all -y

$git –version

$git help

$git help command

CUSTOMIZE GIT ENVIRONMENT:

- Git provides the git config tool, which allows you to set configuration variables. Git stores all global configurations in **.gitconfig** file, which is located in your home directory. To set these configuration values as global, add the --global option.

SYNATAX: $git config --global setting value

$git config --global user.name "Ram"

$git config --global user.email "ram.ashokit@gmail.com"

$git config --global --list

$cat ~/.gitconfig

COLOR HIGHLIGHTING:

- The following commands enable color highlighting for Git in the console.

$git config --global color.ui true
$git config --global color.status auto
$git config --global color.branch auto
$git config --global --list