

# GIT:

- GIT is a version control system that is used to track the changes in the code and helps to collaborate with the developers.
- GIT stands for Global Info tracker.
- GIT is a Distributed Version Control System.
- Helps to track the changes done by the developers while maintaining a history of the changes that are done.

## VERSION CONTROL SYSTEMS:

- Helps to track the changes that are done to file over a period of time.
- 2 types:
  - Centralized VCS(CVCS): Single server that stores all versions(SVN- Subversion).
  - Distributed VCS(DVCS): Each user has a complete copy of the repository(GIT).

## ENVIRONMENTS:

- Developers Environment
- QA Environment
- UAT
- Prod

# GIT ARCHITECTURE:

- Working Directory: The actual files being worked on.
- Staging Area: Temporary area where change sare prepared before committing.
- Local Repository: Our own PC.
- Remote Repository: A repository hosted on a remote server that is GIT.

## GIT STAGES:

- Modified: Change has been made but yet staged.
- Staged: Changes have been added to the staging stage that is ready for commit.
- Commit: Changes are permanently recorded in the Local Repository.

## GIT ACTIONS:

- Push: We are sending the source code from Local Repository to Global Repository.
- Pull: Extraction of the code files/data from the global/remote repository.

# GIT PROTOCOLS:

- HTTPS: HyperText Transfer Protocol Secure.
- SSH: Secure Shell.

## DIFFERENCE BETWEEN GIT AND GITHUB:

<b>GIT</b>	<b>GIT HUB</b>
DVCS used to track the changes in source code.	Cloud based hosting service for GIT repositories.
Manages the code versions locally and collaboratively.	Provides remote storage, collab tools and issue tracking.
Command Line tool.	Web Based tool.
Must be installed on the Local system.	No installation needed, accessed via GITHUB.
Store repositories locally on your computer.	Stores remotely on GITHUB servers.
Requires manual sharing.	Enables real-time collaboration, pull requests, and code reviews.
No authentication.	Requires authentication.
Mostly CLI-based, but GUI tools exist.	Provides a GUI-based interface for managing repositories, pull requests, and issues.
Free and open-source.	Offers free and paid plans.

Version control, branching, merging, reverting changes.	Remote repository hosting, issue tracking, code reviews, project management.
---	--

## DIFFERENCE BETWEEN GITHUB AND GIT LAB:

GITHUB	GIT LAB
A cloud-based platform for hosting Git repositories, collaboration, and CI/CD integration.	A DevOps platform that provides Git repository management, CI/CD, and complete project lifecycle management.
SaaS-based (GitHub.com) and self-hosted via GitHub Enterprise.	Available as SaaS (GitLab.com) and <b>fully self-hosted</b> via GitLab CE (Community Edition) and EE (Enterprise Edition).
Private and public repositories. Free for small teams, with paid plans for more features.	Private and public repositories, with more free-tier features than GitHub.
Requires <b>GitHub Actions</b> or external CI/CD tools like Jenkins, TravisCI, CircleCI.	<b>Built-in CI/CD</b> with GitLab CI/CD pipelines, making automation easier.
Supports SSH keys, personal access tokens, OAuth, and SAML.	Supports SSH keys, LDAP, OAuth, SAML, and Active Directory integration.
Provides Issues, Kanban boards, Milestones, and Discussions.	Provides more advanced project management features, including Epics, Roadmaps.

Uses <b>Pull Requests</b> (PRs) for merging changes.	Uses <b>Merge Requests</b> (MRs) for merging changes.
Acquires GitHub Actions or third-party tools for automation.	Fully integrated <b>CI/CD pipeline</b> with Auto DevOps features.
Widely used by open-source projects and enterprises.	More open-source-friendly, with GitLab CE being <b>fully open-source</b> .
Free for public repositories, paid plans for advanced features.	More free-tier features, paid plans for enterprise support.